SELF-CORRECTING MULTI-CHANNEL BUSSGANG BLIND DECONVOLUTION USING EXPECTATION MAXIMIZATION (EM) ALGORITHM AND FEEDBACK

A Thesis Presented to

The Academic Faculty

by

Sze Ho Tang

In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
Atlanta, Georgia

May 2009

SELF-CORRECTING MULTI-CHANNEL BUSSGANG BLIND DECONVOLUTION USING EXPECTATION MAXIMIZATION (EM) ALGORITHM AND FEEDBACK

Approved by:

Dr. Russell M. Mersereau
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Mark A. Richards
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ye (Geoffrey) Li
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Douglas B. Williams
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Xiaoming Huo
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 3 Nov., 2008

This dissertation is dedicated to my family for their love,

encouragement and support.

ACKNOWLEDGEMENTS

I would like to take this opportunity to convey my deepest gratitude for my advisor, Dr. Russell M. Mersereau, for his continuous support, encouragement, inspiration and kindness during all these years of my pursuing for the degree. I am fortunate to have him as my Ph.D. advisor. He is not only a great mentor but also a wonderful person. Working with him has made this experience a more pleasant one.

I would like to thank my committee members, Dr. Mark A. Richards, Dr. Ye (Geoffrey) Li, and Dr. Douglas B. Williams for their time and valuable suggestions. I would especially like to thank Dr. Mark A. Richards for the valuable discussions on SAR autofocus.

I would also like to thank all the professors and staffs at the Center for Signal and Image Processing (CSIP) for creating an excellent environment for advance learning in the field of digital signal processing.

I am thankful to my close friend, Dr. Clarence Leung, for his friendship and for encouraging me to continue with my pursuit for the degree.

I would like to thank my family for their love, patience, support and the sacrifices they made, in order for me to continue with my pursuing for the degree.

Last but not the least, I would like to thank my parents for their love, for valuing my education, and for the sacrifices they made to make my future a brighter one.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of the research is to design a general blind deconvolution framework that can effectively utilize all available information, tackle severe degradations and be applicable to a wide-range of applications, degradations, signal types and dimensionality with small adaptation. The particular application of greatest interest is the problem of autofocusing in synthetic aperture radar (SAR) and in inverse SAR (ISAR). The motivation comes from the awareness that most of the blind deconvolution schemes available in the literature can only deal with relatively mild degradations [1-8]. This limitation arises from the fact that most schemes cannot easily incorporate all of the information that is available to them. Furthermore, most can guarantee convergence only to a locally optimal solution. For more severe degradations, there are more unknowns and more locally optimal solutions exist; therefore, converging to a globally optimal solution becomes much more difficult. A common remedy to ease the problem is to incorporate known information about the point spread function causing the degradation. Unfortunately, in most cases only very limited information of the point spread function is available, which is not enough to steer the solution to the global optimum.

In this work, we have identified the potential of the Bussgang blind deconvolution framework [1-4] to converge to the globally optimal solution despite its other limitations. As shown by the Benveniste-Goursat-Ruget theorem [2], the Bussgang blind deconvolution framework converges to a globally optimal solution as long as the probability density function of the input signal is non-Gaussian and the support size of the equalization filter size tends to infinity. While it is not possible to have an infinitively long equalization filter support, intuitively, if the signal can be processed in frames instead of in an infinite stream, the support size of the equalization filter should not need to be greater than the frame size. In addition, in the multi-channel case, the attribute of the deconvolution noise on which the Bussgang blind deconvolution framework relies becomes easier to realize. Unlike the requirement of an infinite support size for the equalization filter, the multi-channel implementation is, in fact, both practical and feasible. For example, in the optical imaging case, multiple degraded shots can result from the motion of the targeted subject, or in the synthetic aperture radar (SAR) imaging case, multiple similar flight paths can result in similar, but differently degraded, SAR images. Furthermore, the Bussgang blind deconvolution framework also achieves the goal of having one framework that is applicable to multiple applications. A different application with a different probability density function (pdf) only changes the nonlinearity in the Bussgang blind deconvolution framework. Therefore, we utilize the multi-channel Bussgang blind deconvolution framework as our fundamental building block for the design

of a blind deconvolution procedure that can cope with severe degradations and a wide variety of applications.

To achieve our goal, two obstacles associated with the Bussgang blind deconvolution procedure need to be overcome. These are the requirement that the probability density function (pdf) of the original signal be known and that the original signal be white, which can greatly limit the applicability of the technique.

In this research, we relax the iid requirement and modify the multi-channel Bussgang blind deconvolution framework to allow the pdf of the original signal to be estimated iteratively. We call our proposed modification of the multi-channel Busssgang blind deconvolution framework the self-correcting multi-channel Bussgang (SCMB) blind deconvolution framework. The modifications include a non-conventional feedback mechanism, parameterization of the pdf utilizing a Gaussian mixture model, and parameter estimation using the expectation maximization (EM) algorithm that iterates simultaneously with the original multi-channel Bussgang estimator.

In the dissertation, we demonstrate the effectiveness of the proposed SCMB blind deconvolution framework on two very different problem: the binary image restoration problem and the SAR/ISAR autofocus problem. In the binary image restoration case, our approach recovers severely blurred binary images flawlessly. In the SAR/ISAR autofocus case, our approach outperforms popular autofocus algorithms including phase gradient algorithm (PGA) and minimum entropy autofocus consistently, especially in the ground moving-target ISAR autofocus scenario with both significant translational and rotational motion.

1 Introduction

The blind deconvolution or blind equalization problem seeks to undo the convolution between an input signal $x(\cdot)$ and the impulse response of a linear shift-invariant system $h(\cdot)$ when only the convolved output signal $y(\cdot)$ is available. The term "blind" is used when the impulse response of the system is unknown.

It is assumed that the observed signal $y(\cdot)$ can be modeled as

$$y(\cdot) = h(\cdot) * x(\cdot) + v(\cdot), \tag{1.1}$$

where $*$ denotes the convolution operation and $v(\cdot)$ is additive measurement noise. The arguments of the functions in this equation have been intentionally omitted, since they vary with the application. In some cases the signals are one-dimensional time functions; in others they might be two-dimensional images, for example. A linear system (impulse response) $h(\cdot)$ is adequate for modeling a wide range of physical phenomena including superpositions of signals in multipath communications systems, linear communication channels, and optical degradations such as out-of-focus blurs and motion blurs. In telecommunications applications the problem is often called blind equalization. This is merely terminology; the two problems are the same, although equalization typically requires a "real-time" implementation.

A slightly more complicated model might be needed to account for nonlinear physical phenomena such as channel saturation or the nonlinear sensitivity of film. In these situations, the convolved output signal $y(\cdot)$ with a nonlinearity might be modeled as

$$y(\cdot) = g\{h(\cdot) * x(\cdot)\} + v(\cdot), \tag{1.2}$$

where $g\{\cdot\}$ is a pointwise nonlinearity and $*$ again denotes convolution.

Applications of blind deconvolution arise in a number of diverse fields including communication signal transmission, digital microscopy, reflection seismology in oil exploration, acoustic reverberation cancellation, photographic image restoration and astronomical image restoration and autofocusing in synthetic aperture radars. This latter application is of particular interest in this dissertation.

From equation (1.1), it is quite obvious that if both $h(\cdot)$ and $x(\cdot)$ are completely unconstrained, there will be infinitely many solution pairs for $h(\cdot)$ and $x(\cdot)$ that satisfy the model and yield the same output signal $y(\cdot)$. The same is true for the nonlinear case in (1.2). To limit the space of possible solutions in blind deconvolution or blind equalization,

some constraints on $x(\cdot)$ or $h(\cdot)$ or knowledge of the statistics of the inaccessible input $x(\cdot)$ are assumed. Even when the solution is constrained, however, the solutions may still not be unique. If it is unique, iterative procedures for finding it may converge to a local optimum rather than a global one.

Many blind deconvolution algorithms have been presented in the literature. Some of these procedures are completely general and are not tied to any particular application. Popular blind deconvolution or equalization techniques include, but are not limited to, a maximum likelihood approach implemented using the expectation maximization (EM) algorithm [5, 6] for photographic image restoration, the bispectrum iterative reconstruction algorithm (BIRA) [7] also for photographic image restoration, the tricepstrum equalization algorithm (TEA) [8, 9] for communication channel equalization, Sato's algorithm [10, 11] and Godard's algorithm, both of which are treated as special cases of the Bussgang algorithm [4, 12-17] for communication channel equalization, and finally, stochastic gradient-based blind deconvolution using different figures of merit including minimum entropy [18, 19] and minimum kurtosis [16, 20].

All these algorithms work well when certain assumed conditions are met but, nevertheless, have their limitations when applied to general image restoration. The expectation maximization (EM) algorithm often becomes trapped at a local optimum, especially when the assumed support size of the blur impulse response is large (severe blur); there is no guarantee that the algorithm will converge to the global optimum. Also, the EM algorithm can only work with blurs and input signals that are real because its cost function does not depend upon phase. With the bispectrum iterative reconstruction algorithm (BIRA), the distortion must be one-dimensional. That method also cannot deal with phase distortion. The tricepstrum equalization algorithm (TEA), explicitly assumes that the samples of the unknown are independent and identically distributed (iid). Thus, this algorithm is not applicable to image restoration where this assumption does not hold. With blind equalization using the Bussgang framework, the unknown input signal is again assumed to be iid and its probability density function (pdf) is assumed to be completely known. This is unrealistic for the image restoration problem. Blind image deconvolution utilizing criteria such as minimum entropy or minimum kurtosis assumes that the image blur is a form of lowpass filtering of the original image that results in an increase in the entropy or Gaussianness of the blurred image, so that the pdf of the blurred image will have heavier tails than the pdf of the original image. Thus, minimizing the entropy or kurtosis reverses the blurring process and steers the solution in the right direction. However, there is no inherent stopping mechanism with these criteria. Convergence to the right solution depends solely on the available knowledge of the impulse response $h(\cdot)$ or the availability of additional constraints that can be imposed.

Our goal is to place multiple classes of image restoration (autofocus) problems under one framework, from two-level (binary) images to complex gray-level (SAR) images and for potentially very severe distortion. By understanding the limitations of the algorithms mentioned above, we have decided to base our approach on the Bussgang framework with relaxed constraints. The reasons are the following: (1) Different classes of images can be handled by changing the pdf. (2) The direct inverse filter solution in the Bussgang framework avoids the ill-conditioning problems that are associated with the other methods. This enables it be appropriate for a wider class of distortions. (3) The Benveniste-Goursat-Ruget Theorem [2] guarantees that the algorithm will converge to the global optimum under some mild conditions. This is important for the severely blurred case. (4) Finally, the nonlinearity inherent in the Bussgang framework provides the necessary hook for dealing with the nonlinear blind deconvolution problem as formulated in (1.2), even though that is beyond the scope of this dissertation.

In this dissertation, we propose a self-correcting multi-channel Bussgang blind deconvolution framework that is theoretically applicable to a wide variety of applications. This proposition can be better explained by breaking it down into a number of smaller theses as listed below:

(1) We propose that the independent and identically distribution (iid) requirement on the samples of the input signal with the Bussgang algorithm can be removed.

(2) We propose that the complete knowledge of the probability density function (pdf) of the input signal can be relaxed by simultaneously estimating the pdf and cancelling the pdf estimation error through a feedback structure.

(3) We propose using maximum likelihood implemented using the expectation maximization (EM) algorithm and a Gaussian mixture model to iteratively estimate the pdf parameters of the input signal within the Bussgang framework.

(4) We propose extending the algorithm to the multi-channel case. This improves upon the zero mean, white Gaussian noise requirement on the deconvolution noise and outperforms the single channel case.

(5) We propose applying the self-correcting multi-channel Bussgang blind deconvolution framework on blurred binary images.

(6) We propose applying the self-correcting multi-channel Bussgang blind deconvolution framework to the complex synthetic aperture radar (SAR) autofocus problem.

(7) We propose applying the self-correcting multi-channel Bussgang blind deconvolution framework to the complex inverse synthetic aperture radar (ISAR) autofocus problem.

The remainder of the dissertation is organized as follows: Chapter 2 gives some background on blind deconvolution and techniques that we employ in this thesis. It also provides some background on SAR/ISAR image formation and autofocus. Chapter 3 lays out the self-correcting mult-channel Bussgang (SCMB) blind deconvolution algorithm that we propose. Chapter 4 applies our algorithm to the specific problem of blind binary image restoration, while Chapter 5 applies it to the SAR/ISAR autofocus problem. Chapter 6 concludes the thesis and gives suggestions for future work. Finally, Chapter 0 is an appendix, containing some of the mathematical details that lie behind the algorithm.

2 Background

In this section, we give a brief history of blind deconvolution and blind equalization and state the similarities and differences between the two problems. We divide blind deconvolution and blind equalization approaches into two classes: the impulse response estimation methods and the direct inverse impulse response estimation methods. We then address issues related to each class and their remedies. A representative example for each class will be given and discussed. Finally, some common impulse responses (PSF) grouped by applications are presented, which mark the end of this section.

2.1 Blind deconvolution and equalization

The name blind deconvolution was first used by Stockham [21] for the restoration of old acoustic records in 1975 and the first example of blind equalization was the algorithm proposed by Sato [10] for multilevel amplitude-modulated data transmission in the same year. In both the blind deconvolution and blind equalization problems, the input data is unknown except for its statistics. Only the measurable output and possibly some crude information about the impulse response is accessible depending on the individual algorithm. While blind deconvolution is not tied to a particular application or field, blind equalization usually refers to a data communication application, which is one-dimensional, and where the input data values are assumed to be independent and identically distributed (iid). The iid assumption in data communication is a reasonable one since sequences of independent symbols are typically transmitted in a communication system. Higher order statistics (HOS) approaches, specifically polyspectral-based approaches such as the tricepstrum equalization algorithm (TEA) which relies on the iid assumption explicitly, are not suited to problems like blind image restoration because images rarely have a white spectrum. Furthermore, for HOS-based methods, the complexity increases rapidly with the increase in the dimensionality of the problem. Notwithstanding the similarities and differences between blind deconvolution and blind equalization, we can always classify these approaches into those that estimate the impulse response or the point spread function (PSF) and those that do not.

2.2 Blind deconvolution methods that estimate the impulse response (PSF)

Blind deconvolution or blind equalization algorithms that explicitly solve for the impulse response or the point spread function (PSF) are referred to as impulse response (PSF) estimation-based methods. Some form of inversion of the impulse response is required to

deconvolve the observed signal in order to estimate the unknown input. This inverse process is problematic when the impulse response inversion is ill-posed. An ill-posed condition arises when there are deep nulls in the spectrum of the impulse response. Different classical deconvolution algorithms have different ways of dealing with this ill-posed condition, but most can be viewed as some form of regularization technique that trades fidelity for stability.

2.2.1 The ill-posed condition

The mathematical term "well-posed problem" stems from a definition given by Jacques Hadamard in 1902. He believed that mathematical models of physical phenomena should have the properties that: 1) A solution exists, 2) The solution is unique. And 3) the solution depends continuously on the data in some reasonable topology and, therefore, is well-posed. However, the backward (inverse) problem of such a well-posed problem may well be ill-posed.

For example, a data communication channel can be modeled using the linear model (1.1) in one dimension and written as

$$y(n) = h(n) * x(n) + v(n), \tag{2.1}$$

where $*$ denotes convolution, $h(n)$, $x(n)$, $v(n)$ and $y(n)$ are respectively the channel point spread function (PSF), input data, channel additive noise and channel output data. This linear communication channel model (2.1) can also be written in matrix-vector form as,

$$\mathbf{y} = \mathbf{H}\,\mathbf{x} + \mathbf{v}, \tag{2.2}$$

where $\mathbf{y}$, $\mathbf{x}$, $\mathbf{v}$ are $N\times 1$ vectors and $\mathbf{H}$ is an $N\times N$ Toeplitz matrix. Here N is the data length.

$$\mathbf{y} = \begin{pmatrix} y(0) \\ y(1) \\ \vdots \\ \vdots \\ y(N-1) \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x(0) \\ x(1) \\ \vdots \\ \vdots \\ x(N-1) \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} v(0) \\ v(1) \\ \vdots \\ \vdots \\ v(N-1) \end{pmatrix}, \quad \text{and}$$

$$\mathbf{H} = \begin{pmatrix} h(0) & 0 & 0 & 0 & \cdots & 0 & 0 \\ h(1) & h(0) & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & h(M-1) & \cdots & h(1) & h(0) & 0 \\ 0 & \cdots & 0 & h(M-1) & \cdots & h(1) & h(0) \end{pmatrix}.$$

Note that M is the length of the channel point spread function (PSF) and $\mathbf{H}$ becomes a block-Toeplitz matrix in the two-dimensional case.

The physical phenomenon that yields the channel output vector $\mathbf{y}$ from the inputs $\mathbf{x}$ and $\mathbf{v}$ is a well-posed problem. However, the inverse problem of solving for the channel input vector $\mathbf{x}$ is often ill-posed. That problem can be stated as

$$\mathbf{x} = \mathbf{H}^{-1}\mathbf{y} - \mathbf{H}^{-1}\mathbf{v}. \tag{2.3}$$

Depending on the properties of the inverse matrix $\mathbf{H}^{-1}$, a bounded perturbation introduced by the noise $\mathbf{v}$ can result in an unbounded perturbation in the solution $\mathbf{x}$. The degree of ill-posedness can be measured by the condition number of $\mathbf{H}$; the higher the condition number, the more ill-posed the inverse problem. When the l_2-norm is used, a common definition for the condition number of $\mathbf{H}$, $\kappa(\mathbf{H})$, is

$$\kappa(\mathbf{H}) = \frac{\sigma_{\max}(\mathbf{H})}{\sigma_{\min}(\mathbf{H})} = \sqrt{\frac{\lambda_{\max}(\mathbf{H}^H\mathbf{H})}{\lambda_{\min}(\mathbf{H}^H\mathbf{H})}}, \tag{2.4}$$

where $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest singular values, $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalues and the superscript H denotes the Hermitian transpose of a matrix.

The degree of ill-posedness can also be measured in terms of the magnitude spectrum of the impulse response $h(n)$ when its support is small compared to the data length, $M << N$. In this case, the Toeplitz matrix $\mathbf{H}$ can be approximated by a circulant matrix, so that

$$\mathbf{F}\,\mathbf{H}\,\mathbf{F}^H = \mathbf{D}, \tag{2.5}$$

where $\mathbf{F}$ is the $N \times N$ one-dimensional discrete Fourier transform (DFT) kernel and $\mathbf{D}$ is a diagonal matrix with the diagonal entries equal to the DFT coefficients of the impulse response $h(n)$. By recognizing $\mathbf{F}$ as the matrix of eigenvectors of $\mathbf{H}$ and $\mathbf{D}$ as the diagonal matrix of eigenvalues of $\mathbf{H}$, it is clear that the condition number can be evaluated as

$$\kappa(\mathbf{H}) = \frac{\max\{|DFT_N(h(n))|\}}{\min\{|DFT_N(h(n))|\}}, \tag{2.6}$$

where the DFT can be evaluated efficiently using the fast Fourier transform (FFT) algorithm. Thus, it can be observed that the whiter the magnitude spectrum of the impulse response, the smaller the condition number and the more well-posed the inverse problem becomes.

2.2.2 Regularization techniques

Regularization is a well-studied method for solving ill-posed problems. It turns an ill-posed problem into a well-posed one by sacrificing some fidelity of the solution for stability (Hadamard's third condition). Incorporating prior knowledge about the input data is the main idea behind regularization. Depending on how the prior knowledge is being utilized, there are two types of regularization methods: deterministic regularization techniques and stochastic regularization techniques.

Deterministic regularization techniques limit or regulate the possible solution space using deterministic constraints. One deterministic constraint that is often used in image restoration is a smoothness constraint, which imposes the prior knowledge that the original image is dominated by its low-frequency content. In the classical deconvolution problem where the impulse response is assumed to be known, deterministic constraints like a smoothness constraint are often included as a Lagrange multiplier as shown below

$$\hat{\mathbf{x}} = \arg\{\min_{\mathbf{x}}\{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + k\|\mathbf{C}\mathbf{x}\|^2\}\}. \tag{2.7}$$

Here k is the Lagrange multiplier, or regularization parameter, and $\mathbf{C}$ is a regularization operator. One example of a non-optimal choice of the regularization operator $\mathbf{C}$ for the image restoration problem is the discrete Laplacian operator which is the sum of the second spatial derivatives in the horizontal and vertical directions. The larger the regularization parameter k, the smoother the solution becomes.

Stochastic regularization techniques limit or regulate the possible solution space using a stochastic model of the input data. Like its deterministic counterpart, the stochastic model constrains the feasible solution to a smaller set of candidates. In most cases, a stochastic model of the input data is not available and must be estimated from the output data or from prototypical training data. One example of such a stochastic model commonly used in image restoration is an auto-regressive (AR) model

$$\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{w}\,, \tag{2.8}$$

where $\mathbf{A}$ is a block-Toeplitz matrix representing a two-dimensional linear shift invariant filtering operation and $\mathbf{w}$ is a white Gaussian noise vector. The matrix $(\mathbf{I}-\mathbf{A})$ can be interpreted as a regularization operator while the inverse of the variance of $\mathbf{w}$, σ_w^{-2}, can be interpreted as the regularization parameter. The larger the variance of $\mathbf{w}$, the AR modeling error, the smaller the regularization parameter and, therefore, the less effective is the constraining effect of the AR model (regularization operator).

2.2.3 The expectation maximization (EM) algorithm

The expectation maximization (EM) algorithm is a popular approach for solving the blind deconvoluton problem which belongs to the category of impulse response or PSF estimation-based methods. The impulse response $h(\cdot)$ is iteratively and explicitly estimated (together with other parameters). Each iteration consists of two steps. At the k-th iteration, the expectation step (E-step) estimates the input data $x(\cdot)^{(k)}$ or $\mathbf{x}^{(k)}$, and the maximization step (M-step) estimates the parameter set $\phi^{(k)}$ which includes the impulse response $h(\cdot)^{(k)}$, the noise variance $\sigma_v^{2\,(k)}$ and possibly other parameters. In the E-step, the conditional mean of the input data is estimated using the parameter set from the previous iteration $\mathbf{x}^{(k)} = E\{\mathbf{x}|\mathbf{y};\phi^{(k-1)}\}$; in the M-step this estimate is used to update the parameter set.

The expectation maximization (EM) algorithm is an iterative computation of the maximum-likelihood (ML) estimate of $x(\cdot)$. The output data $\mathbf{y}$ is viewed as incomplete data that is related to some complete data $\mathbf{z}$ through a non-invertible many-to-one transformation $T\{\cdot\}$, $\mathbf{y} = T\{\mathbf{z}\}$. One possible choice of the complete data is a concatenation of the input and output data, $\mathbf{z} = [\mathbf{x}^T\ \mathbf{y}^T]^T$. The EM algorithm iteratively chooses the parameter set ϕ so as to maximize the logarithm of the likelihood (probability density) function of the complete data, $\log(p(\mathbf{z};\phi))$. However, since the complete data is not accessible, the conditional expectation of the log-likelihood function $Q(\phi\,|\,\phi^{(k-1)})$ is maximized instead, as shown below.

$$Q(\phi\,|\,\phi^{(k-1)}) = E\{\log(p(\mathbf{z};\phi))\,|\,\mathbf{y};\phi^{(k-1)}\}\,, \tag{2.9}$$

and

$$\phi^{(k)} = \arg\{\max_{\phi}\{Q(\phi\,|\,\phi^{(k-1)})\}\}\,. \tag{2.10}$$

Convergence of the EM algorithm is guaranteed if $Q(\phi\,|\,\phi^{(k-1)})$ is continuous with respect to both ϕ and $\phi^{(k-1)}$ at each iteration. However, it does not necessarily converge to the

global maximum. The larger the number of parameters in the parameter set ϕ, the more likely the algorithm is to converge to a local maximum. In the blind image restoration problem, constraints such as the conservation of energy, symmetry and smoothness are commonly used to reduce number of parameters and hence the likelihood of converging to a local maximum.

When the probability density function (pdf) of the complete data $\mathbf{z}$ has a multivariate Gaussian distribution, the EM algorithm has a closed-form solution. Therefore, no numerical optimization technique of any kind is required. In this case, the algorithm becomes extremely efficient. The complete data $\mathbf{z}$ has a multivariate Gaussian distribution if the following conditions hold: 1) The noise term of the blind deconvolution model $\mathbf{v}$ in (2.2) is white and Gaussian. 2) The complete data is chosen as $\mathbf{z} = [\mathbf{x}^T \mathbf{y}^T]^T$. 3) The input data model in (2.8) is employed for which the input data modeling noise $\mathbf{w}$ is white and Gaussian.

2.3 Direct inverse impulse response estimation-based blind deconvolution

Blind deconvolution or blind equalization algorithms that solve for the inverse impulse response or equalization filter directly are referred to as direct inverse impulse response estimation-based methods. Unlike the impulse response or PSF estimation-based methods, no inversion of an estimated impulse response or PSF is involved. Therefore regularization techniques that limit ill-posedness by trading fidelity for stability, both of which arise from the inverse problem, can be avoided. Algorithms belonging to this category include stochastic gradient-based methods with different figures of merit such as minimum entropy [18, 19] and minimum kurtosis [16, 20]. Sato's algorithm [10, 11] and Godard's algorithm, under the umbrella of the Bussgang algorithm [1, 12, 15] for communication signal equalization, also belong to this category.

2.3.1 Adaptive filtering

Adaptive filtering is a recursive procedure for estimating the filter coefficients so that the filter's output signal matches the known desired signal. Unlike the optimal Wiener filter, it is possible for the adaptive filter to perform this task without knowing the statistical characteristics of the input data. An adaptive filtering algorithm starts with an initial guess of the filter coefficients and iteratively converges to the optimal Wiener solution in a statistical sense. For a streaming input signal with nonstationary statistics, adaptive filtering acts as a tracker. The Bussgang technique for streaming input data can be viewed as an

extension of the adaptive filtering technique with the desired signal generated by a nonlinearity function that utilizes available statistical characteristics of the input data.

2.3.2 Bussgang technique

The Bussgang technique is a member of the class of nonlinear blind equalization algorithms. It is a very popular blind equalization framework that iteratively and directly estimates the inverse impulse response. Different choices of the cost function for the Bayes' risk result in a different Bussgang zero-memory nonlinearity. Popular algorithms for the data communication blind equalization problem including Sato's algorithm [10], Godard's algorithm [22] and the constant modulus algorithm (CMA) [23-25], all of which result from different choices of the cost function. Thus, they can all be viewed as special cases of the Bussgang blind equalization algorithm.

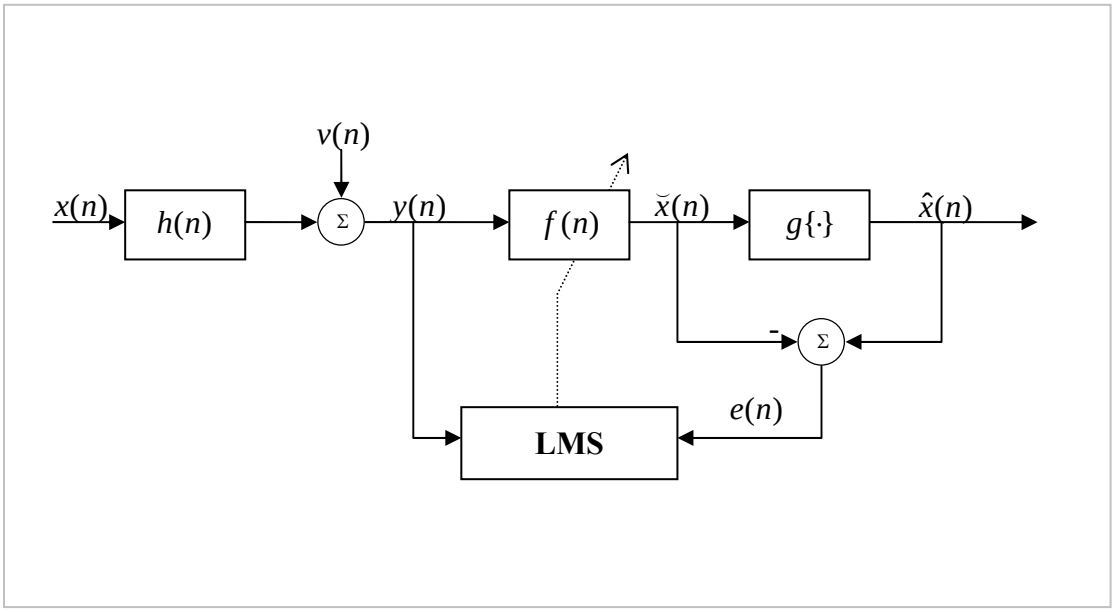


Figure 2-1 Functional block diagram of Bussgang blind equalization

Figure 2-1 shows a general block diagram of Bussgang blind equalization where the nomenclature is as follows:

$x(n)$	- Unknown input data.
$h(n)$	- Unknown impulse response or point spread function (PSF).
$v(n)$	- Additive noise.
$y(n)$	- Measurable output data.
$f(n)$	- Inverse filter impulse response or equalization filter.
$\breve{x}(n)$	- Deconvolved output.

$g\{\cdot\}$ - Zero-memory nonlinearity.

$\hat{x}(n)$ - Output of the Bussgang algorithm (estimate of $x(n)$)
(the desired signal in the context of adaptive filtering).

$e(n)$ - Estimation error (in the context of adaptive filtering).

LMS - Least mean square adaptive algorithm.

The goal of the Bussgang blind equalization algorithm is to match the distribution (pdf) of the output of the Bussgang algorithm (the recovered input signal), $\hat{x}(n)$, with the pdf of the unknown input signal $x(n)$. The name "Bussgang" comes from the fact that the procedure converges in the mean to the desired result when the deconvolved output $\breve{x}(n)$ is a so-called Bussgang process. A Bussgang process is a stochastic process whose autocorrelation is equal to its cross-correlation with the output of the zero-memory nonlinearity $g\{\cdot\}$ as shown below:

$$E\{\breve{x}(n)\breve{x}(n+m)\} = E\{\breve{x}(n)g\{\breve{x}(n+m)\}\}. \tag{2.11}$$

Bussgang blind deconvolution can also be viewed as an adaptive filtering [26] problem which in the context of adaptive filtering theory has an unknown desired output. The absence of the desired output is addressed by a zero-memory nonlinearity designed to estimate the signal which most closely matches the distribution of the input signal.

Bussgang blind equalization assumes and requires 1) that the input signal samples $x(n)$ are independent and identically distributed (iid) with a white power spectrum and 2) that pdf of those samples is known and non-Gaussian. In the data communication application, since a sequence of independent symbols is usually transmitted, the iid assumption is generally appropriate. Furthermore, a non-Gaussian distribution, such as a uniform distribution, usually models the input sequence in that application quite well. However, these restrictions severely limit the applicability of the Bussgang algorithm in other applications.

2.3.2.1 Nonlinearity

The zero-memory nonlinearity $g\{\cdot\}$ in the Bussgang algorithm results from the Bayesian estimation of the unknown input data. It depends upon 1) the non-Gaussian probability density function (pdf) of the unknown input data, 2) the deconvolution noise, $w(n)$, having a zero mean, white Gaussian distribution, and 3) the choice of the cost functional $C(\cdot)$ for the Bayes risk $J(\hat{x})$. The expression for the Bayes risk is given by

$$J(\hat{x}) = \iint C(x - \hat{x}(\breve{x};\Phi))\, p_{x,\breve{x}}(x,\breve{x})\, dx\, d\breve{x}, \tag{2.12}$$

Here Φ is the vector containing the parameters that model the pdf of the unknown input data and the variance of the deconvolution noise, σ_w^2. The output of the Bussgang algorithm, which minimizes the Bayes risk, is

$$\hat{x} = g\{\breve{x};\Phi\} = \arg\left\{\min_{\hat{x}}\left\{J\left(\hat{x}\left(\breve{x};\Phi\right)\right)\right\}\right\}, \tag{2.13}$$

where x, $\breve{x}$ and $\hat{x}$ are respectively the random variables of the unknown input data, the nonlinearity input and the nonlinearity output $x(n)$, $\breve{x}(n)$ and $\hat{x}(n)$. $C(\cdot)$ is the cost functional for the Bayes risk, $J(\hat{x})$, and $p_{x,\breve{x}}(x,\breve{x})$ is the joint probability density function of x and $\breve{x}$. If the cost functional is chosen to be the l_2-norm, that is, $C(\cdot) = \|\cdot\|_2$ then Bayesian estimation becomes identical to minimum mean square error (MMSE) estimation.

The deconvolution noise $w(n)$ in the context of Bussgang blind deconvolution is defined as the output of the equalization filter $\breve{x}(n)$ minus the unknown input data $x(n)$, that is, $w(n) = \breve{x}(n) - x(n)$. The pdf of the deconvolution noise $w(n)$ is used in deriving the nonlinearity $g\{\cdot\}$ of the Bussgang algorithm. The requirement of the input data samples to be independent and identically distributed ensures that the deconvolution noise will be zero mean, white Gaussian noise. Thus, the only free parameter associated with $w(n)$ is its variance.

The adaptation of the equalization filter $f(n)$ can be achieved using any of a variety of well-studied approaches including the steepest descent algorithm, the stochastic gradient algorithm from adaptive filtering (LMS, NLMS, RLS) or Wiener-based algorithms [27]. LMS is particularly popular for a one-dimensional signal stream because of its simplicity and its low memory requirement, since no correlation matrix or matrix inverse calculation is needed.

In 1980 Benveniste [2] derived sufficient conditions for the global convergence of the Bussgang algorithm for an infinite length equalizer. The Benveniste-Goursat-Ruget theorem states that global convergence of the Bussgang algorithm is guaranteed if the pdf of the unknown input data sequence $x(n)$ is sub-Gaussian distributed (shorter tail than the Gaussian distribution) and the second derivative of the estimation error $e(n)$, $e''(n)$, as defined in Figure 2-1 is negative. As shown in the pdf

$$p(x) = K\, e^{-\left|\frac{x}{\beta}\right|^{v}}, \tag{2.14}$$

for $v > 2$, x is sub-Gaussian distributed and for $v \to \infty$, x is uniform distributed.

2.4 Gaussian mixture estimation using the EM algorithm

Assume that we have multidimensional data, $\mathbf{z}$, that can be modeled by a multivariate Gaussian mixture model, as shown below:

$$p_{\mathbf{z}}(\mathbf{z}) = \sum_{c=1}^{C} p_{\mathbf{z}/\mathbf{x}_c}(\mathbf{z}/\mathbf{x}_c)\, p_{\mathbf{x}_c}(\mathbf{x}_c), \tag{2.15}$$

where $\mathbf{x}$ is a hidden variable and $\mathbf{x} = \mathbf{x}_c$ when the data point $\mathbf{z}$ is generated by the c^{th} component of the mixture. The conditional probability density function, $p_{\mathbf{z}/\mathbf{x}_c}(\mathbf{z}/\mathbf{x}_c)$, is the probability of the data point $\mathbf{z}$ given that it is generated from the c^{th} Gaussian component. It has a Gaussian form. Thus,

$$p_{\mathbf{z}/\mathbf{x}_c}(\mathbf{z}/\mathbf{x}_c) = \left|2\pi\Lambda_{\mathbf{x}_c}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}\left(\mathbf{z}-\boldsymbol{\mu}_{\mathbf{x}_c}\right)^T \Lambda_{\mathbf{x}_c}^{-1}\left(\mathbf{z}-\boldsymbol{\mu}_{\mathbf{x}_c}\right)}, \tag{2.16}$$

where $\boldsymbol{\mu}_{\mathbf{x}_c}$ and $\Lambda_{\mathbf{x}_c}$ are, respectively, the mean vector and the covariance matrix for the c^{th} component. Moreover, the weights (probabilities) of the C components add to one, so that

$$w_c \triangleq p_{\mathbf{x}_c}(\mathbf{x}_c), \tag{2.17}$$

and

$$\sum_{c=1}^{C} w_c = \sum_{c=1}^{C} p_{\mathbf{x}_c}(\mathbf{x}_c) = 1. \tag{2.18}$$

Therefore (2.15) can be written as

$$p_{\mathbf{z}}(\mathbf{z}) = \sum_{c=1}^{C} w_c \; p_{\mathbf{z}/\mathbf{x}_c}(\mathbf{z}/\mathbf{x}_c)\ . \tag{2.19}$$

The expectation maximization (EM) algorithm, as described in Section 2.2.3, can be used to estimate the parameters of the Gaussian mixture. Using the terminology of the EM algorithm, the available data $\mathbf{z}$ is the incomplete data and together with the hidden data $\mathbf{x}$ forms the so called complete data. The parameter set in this case is $\phi = \left\{w_c, \boldsymbol{\mu}_{\mathbf{x}_c}, \boldsymbol{\Lambda}_{\mathbf{x}_c}\right\},\ \forall c \in [1, C]$. By taking the derivative of the conditional expectation of the log-likelihood function $Q(\phi \mid \phi^{(k)})$, as defined in (2.9), with respect to the parameter set, we generate the update equations for the E-step and the M-step as follows:

E-step:

$$y(i)^{(k)} = w_c{}^{(k)} \left|2\pi \boldsymbol{\Lambda}_c{}^{(k)}\right|^{-\frac{1}{2}} e^{-\frac{1}{2}\left(\mathbf{z}(i)-\boldsymbol{\mu}_c{}^{(k)}\right)^T \boldsymbol{\Lambda}_c{}^{(k)\,-1}\left(\mathbf{z}(i)-\boldsymbol{\mu}_c{}^{(k)}\right)} \tag{2.20}$$

M-step:

$$w_c{}^{(k+1)} = \frac{1}{N}\sum_{i=1}^{N} y(i)^{(k)} \tag{2.21}$$

$$\boldsymbol{\mu}_c{}^{(k+1)} = \frac{\sum_{i=1}^{N} \mathbf{z}(i)\; y(i)^{(k)}}{\sum_{i=1}^{N} y(i)^{(k)}} \tag{2.22}$$

$$\boldsymbol{\Lambda}_c{}^{(k+1)} = \frac{\sum_{i=1}^{N} \left(\mathbf{z}(i)-\boldsymbol{\mu}_c\right)\left(\mathbf{z}(i)-\boldsymbol{\mu}_c\right)^T y(i)^{(k)}}{\sum_{i=1}^{N} y(i)^{(k)}}, \tag{2.23}$$

where k is the iteration number and N is the number of available multidimensional data points $\mathbf{z}$.

2.5 Synthetic/Inverse synthetic aperture radar (SAR/ISAR)

In a conventional radar, resolution in the azimuth (cross-range) direction improves as the antenna aperture or the radar central frequency increases, and resolution in the range direction improves as the radar signal bandwidth increases. A radar in motion can combine echoes from multiple pulses transmitted at different spatial locations to improve azimuth resolution. This signal synthesis can achieve an azimuth resolution that would require a much larger antenna aperture in a conventional radar. This concept is called synthetic aperture radar (SAR) [28, 29]. Alternatively, instead of having the radar moving and the target stationary, as in the SAR mode, the inverse synthetic aperture radar (ISAR) mode operates with the target moving and the radar stationary. While the signals are similar in both cases, image formation in the ISAR mode is more challenging simply because the motion of the target is not known. The motion of the radar platform in SAR mode can be estimated quite well when an inertial measurement unit (IMU) and/or global positioning system (GPS) are available.

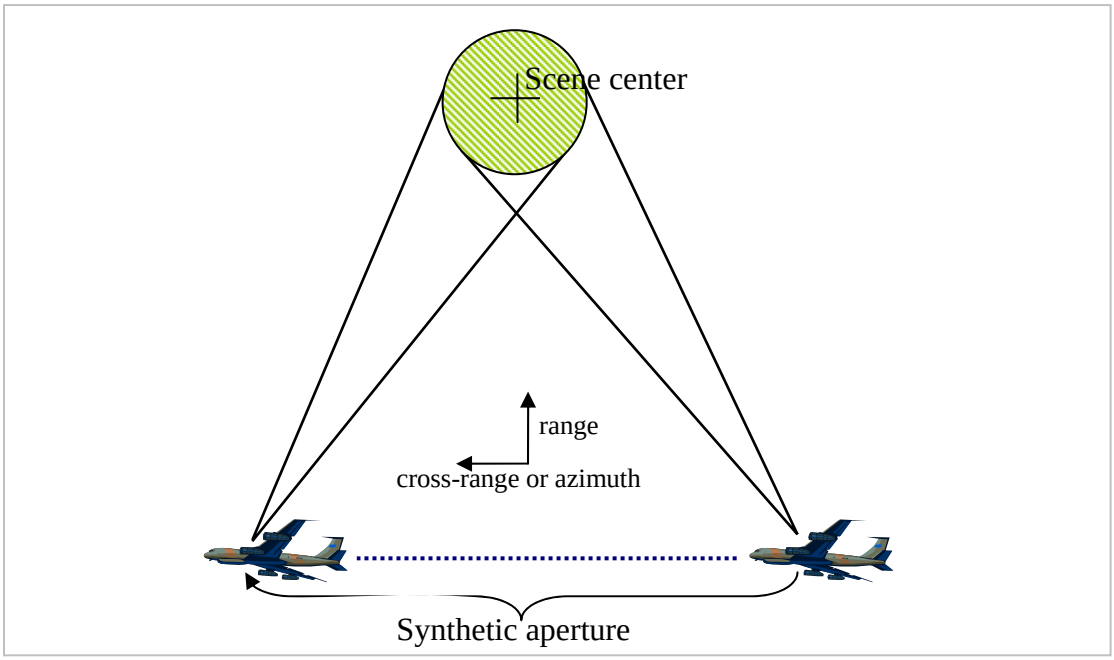


Figure 2-2 Synthetic aperture radar (SAR) in spotlight mode

Figure 2-2 shows the operation of a synthetic aperture radar (SAR) in spotlight mode. The size of the synthetic aperture increases with a longer flight path. The larger the synthetic aperture, the finer the azimuth resolution becomes. In spotlight mode, the radar constantly points to the scene center from pulse to pulse. The scene center is the center of the SAR image that is formed.

2.5.1 The SAR image formation process (IFP)

SAR image formation consists of the processing operations required to produce an image from a SAR signal and flight path information. This data processing includes IMU and GPS data processing to extract the flight path information, matched filtering and demodulation processing to obtain the baseband in-phase and quadrature phase (IQ) signals, motion compensation to ensure that the signal is referenced to the desired scene center correctly and that the coherence between echoes is correct, image formation from the IQ signal (for example using the polar format algorithm (PFA) or back projection algorithm), and SAR autofocus to compensate for motion error from the IMU and GPS, which is more apparent for high frequency band radars [28, 29].

2.5.2 SAR autofocus

SAR autofocus corrects the defocused SAR image by compensating for phase errors from pulse to pulse. This phase error can be created by uncompensated motion between the

SAR antenna phase center (APC) and the scene being imaged, algorithm approximations, hardware limitations and jitter in the demodulation timing. For high resolution SAR imagery, tight tolerances are required for flight platform positioning errors which may not be achievable even with today's technology. Furthermore, some sources of phase error such as atmospheric turbulence can cause the demodulation timing error to become independent of the precision of the onboard IMU and GPS. The use of autofocus techniques in SAR imaging eliminates the significant hardware cost associated with the ultra-high-accuracy navigation system needed to keep the flight platform positioning error down. At the same time, it can also take care of other sources of defocusing error. As a result, SAR autofocus is an indispensable part of the SAR image formation process.

The phase gradient algorithm (PGA) [29-31] is the most popular and widely utilized algorithm for SAR autofocus. Unlike other SAR autofocus approaches, which depend upon a low-order phase error model such as a quadratic or cubic phase error model, PGA does not use any phase error model. Thus, it can deal with an arbitrarily high-order phase error. The theory behind the PGA uses a range profile model in the range-compression (range-azimuth, range-pulse or range-doppler) domain. It makes the following assumptions: (1) the range profiles of two adjacent pulses are the same; (2) the phase error is range invariant or pulse dependent only; (3) white additive complex Gaussian noise adequately models the range profile noise. Apart from these, in the algorithm a circular shift in the cross-range dimension of the SAR image is performed so that the strongest scatterer at each range gate is aligned at the center. Under these conditions, we have,

$$\begin{aligned} \bar{g}(r,p) &= a(r)e^{i\phi(p)} + n(r,p) \\ \bar{g}(r,p+1) &= a(r)e^{i\phi(p+1)} + n(r,p+1) \end{aligned} \tag{2.24}$$

where r and p are, respectively, the range and pulse indexes, $a(r)$ is the ideal range profile, $\phi(p)$ is the phase error for pulse p, $n(r,p)$ is two-dimensional white complex Gaussian noise and $\bar{g}(r,p)$ is the actual measured range profile. Define

$$\mathbf{x} = \begin{bmatrix} \bar{g}(r,p) \\ \bar{g}(r,p+1) \end{bmatrix} \quad \text{and} \quad \Delta\phi = \phi(p+1) - \phi(p). \tag{2.25}$$

We see that $\mathbf{x}$ has a zero-mean bivariate complex Gaussian distribution. By maximizing the log-likelihood function, $\ln\left(p_{\mathbf{x}}(x;\Delta\phi)\right)$, with respect to the phase gradient, $\Delta\phi$, we have

$$\Delta\hat{\phi}_{ML}(p+1) = \angle\left[\sum_{r=1}^{N} \bar{g}(r,p)^{*}\,\bar{g}(r,p+1)\right]. \tag{2.26}$$

Correction of the phase error using the phase gradient $\Delta\hat{\phi}_{ML}(p)$ can focus the SAR image but a constant phase error across all pulses can still exist. This can cause a shift in the SAR image. PGA autofocus works well if the error that causes the defocusing can be adequately modeled by pulse dependent phase errors across pulses only. In general, this condition does not apply in the ISAR autofocus case.

Minimum entropy autofocus [19, 32, 33] assumes that the entropy of a focused SAR image is lower than the entropy of a defocused one. This is equivalent to the assumption that the spikiness of a focused SAR image is higher than the spikiness of a defocused one, since minimizing the entropy is equivalent to increasing the spikiness of a SAR image. Based on this reasoning, minimizing the entropy is expected to bring the SAR image in focus. Minimum entropy autofocus is applied in the complex image space, where the entropy H is defined as follows:

$$H = \sum_{r,c} p(r,c) \ln\left[\frac{1}{p(r,c)}\right], \tag{2.27}$$

and

$$p(r,c) = \frac{|x(r,c)|^2}{\sum_{r,c} |x(r,c)|^2}. \tag{2.28}$$

In these expressions, r and c are, respectively, the range and cross-range indexes of a pixel in the image, $x(r,c)$ is the complex intensity of a pixel, and $p(r,c)$ is the normalized power of a pixel. The normalization is such that the power of all pixels in the image sums to one. As a result, the normalized power, $p(r,c)$, has the attributes of a probability associated with pixel (r,c). Unlike PGA, where the calculation of the phase gradient of each pulse depends only on two adjacent pulses as shown in equation (2.26), with minimum entropy autofocus the phase corrections along the entire cross-range data set have to be found simultaneously. Because there can be hundreds to thousands of phase corrections (unknowns) which have to be solved simultaneously depending on the cross-range dimension of the SAR image, the method is often not feasible. One remedy that is frequently implemented is to fit the phase corrections with a low-order polynomial or basis functions so as to drastically reduce the number of unknowns as shown below

$$\phi\left(\vec{b}, k_c\right) = \sum_{i=0}^{N} b(i) k_c^{\,i}, \tag{2.29}$$

where N is the order of the polynomial, $b(i)$ and $\vec{b}$ are respectively the $i-th$ coefficient and the coefficient vector of the polynomial, and k_c is the index of the k-space in the cross-range dimension. The coefficient vector $\vec{b}_{me}$ obtained by minimizing the entropy of the SAR image $H(x)$ becomes

$$\vec{b}_{me} = \arg\left\{ \min_{\vec{b}} \left\{ H\left(\mathrm{F}^{-1}\left\{ X(k_r, k_c) e^{j\phi(\vec{b}, k_c)} \right\} \right) \right\} \right\}, \tag{2.30}$$

where F^{-1} is the two-dimensional inverse Fourier transform and $X(k_r, k_c)$ is the IQ signal on the k-space rectangular grid.

2.5.3 ISAR autofocus

ISAR autofocus [34, 35] forms a focused SAR image of a moving target instead of a static scene as in the SAR case. On the surface, the ISAR autofocus problem should be very similar to the SAR autofocus problem since it is only the relative motion between the radar and the target (scene) that matters and it doesn't depend on which one is actually moving. What makes the ISAR autofocus problem more difficult than the SAR autofocus problem is the uncertainty of both the translational and rotational motion of the target, which results in uncertainty of the azimuth angle with respect to the target body axis for each radar pulse. The azimuth error is depicted in Figure 2-3, where the red line is the actual line of sight (LOS), the blue line is the estimated line of sight, and the azimuth error is the angle difference between the red and the blue line for each pulse.

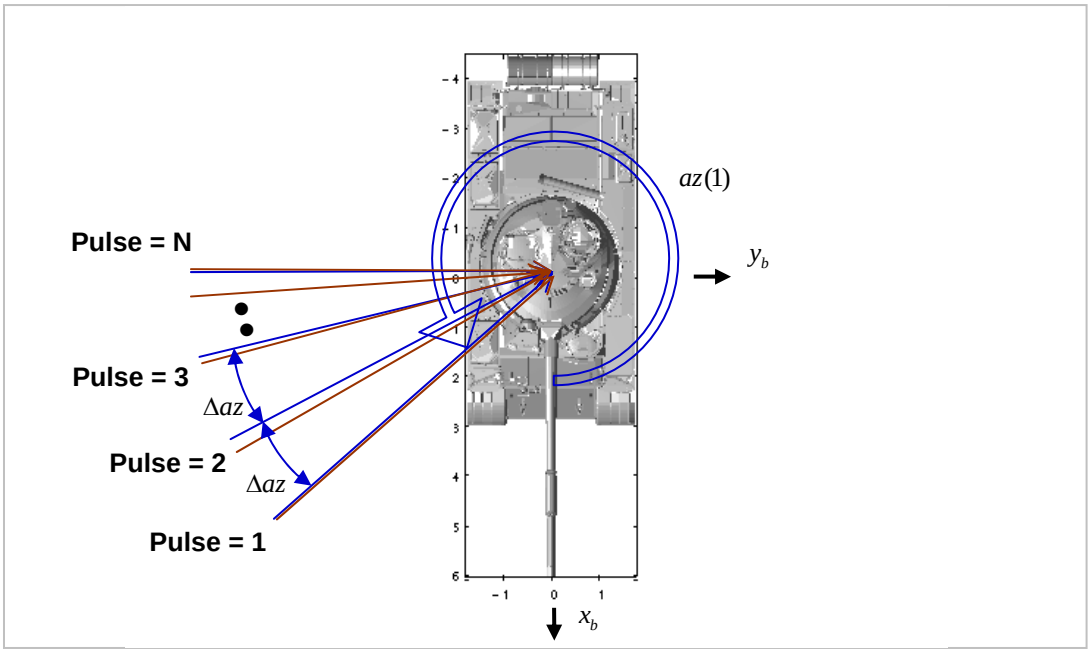


Figure 2-3 Illustration of azimuth errors w.r.t. the target body axis from pulse to pulse (the red line is the actual LOS and the blue line is the estimated LOS).

This means that, in general, the ISAR autofocus problem not only has to compensate for the range errors (phase error) as in the SAR autofocus problem, but it also has to compensate for this azimuth error. Depending on the complexity and characteristics of the target motion within the dwell time, the azimuth error corresponding to a constant speed linear target motion might be no more than a simple azimuth scaling, which should not create any defocusing other than a scaling error of the ISAR image itself. In general, however, ISAR autofocus requires a two-dimensional phase and amplitude error compensation.

2.6 Characteristic of the point spread function (PSF)

In the blind deconvolution or blind equalization problem, the point spread function (PSF) or impulse response of the degradation process is estimated either directly or implicitly, but nevertheless simultaneously, with the unknown input data. Different applications naturally have different point spread functions associated with the underlying physical process. Here, we list a few of the common ones encountered in the fields of data communication, photographic imaging and synthetic aperture radar (SAR) imaging.

2.6.1 PSF in data communication

Data communication generally deals with complex signals, where both the transmitted data sequence and the point spread function are complex. An example of a complex communication signal is a quadrature amplitude modulated (QAM) signal. The point spread function describes the communication channel dispersion, which is a one-dimensional signal that is usually non-minimum phase and is ill-posed. This channel dispersion is typically a lowpass process that satisfies the conservation of energy as shown below

$$\sum_{n \in S_h} |h(n)|^2 = 1, \tag{2.31}$$

where S_h is the support of the point spread function $h(n)$. Conservation of energy here means if the channel has a white noise input, channel dispersion that satisfies equation (2.31), guarantees the channel output is also white noise with the same energy.

2.6.2 PSFs in photographic imaging

In photographic imaging, common degradations are due to out-of-focus, diffusion and motion. Out-of-focus degradations result from focusing to the wrong depth. Diffusion degradation is caused by imperfection of the lens or by atmospheric turbulance. Finally, motion degradation is the result of relative motion between the camera and the subject in the picture during the integration time of the sensor. The point spread functions or impulse responses of out-of-focus, diffusion and motion are all two-dimensional, real signals satisfying the unity D.C. gain constraint given by

$$\sum_{(m,n) \in S_h} h(m,n) = 1. \tag{2.32}$$

The impulse response of out-of-focus blur is often modeled as a pillbox-shaped function,

$$h(m,n) = \begin{cases} \frac{1}{k} & if \quad m^2 + n^2 < r^2 \\ 0 & elsewhere \end{cases}. \tag{2.33}$$

The impulse response of diffusion blur, is often modeled as a Gaussian blur,

$$h(m,n) = \left(\frac{1}{k}\right) e^{-\left(\frac{m^2+n^2}{2\sigma^2}\right)}, \tag{2.34}$$

and the impulse response of motion blur with an angle θ, a directional blur is,

$$h(m,n)=\begin{cases}\dfrac{1}{k} & if \quad m^2+n^2<r^2 \;\;\&\;\; n=integer\{\tan(\theta)\,m\} \\ 0 & elsewhere\end{cases}. \tag{2.35}$$

In these expressions k is a normalizing factor so that the unity D.C. gain in (2.32) is satisfied and r is the radius of the blur.

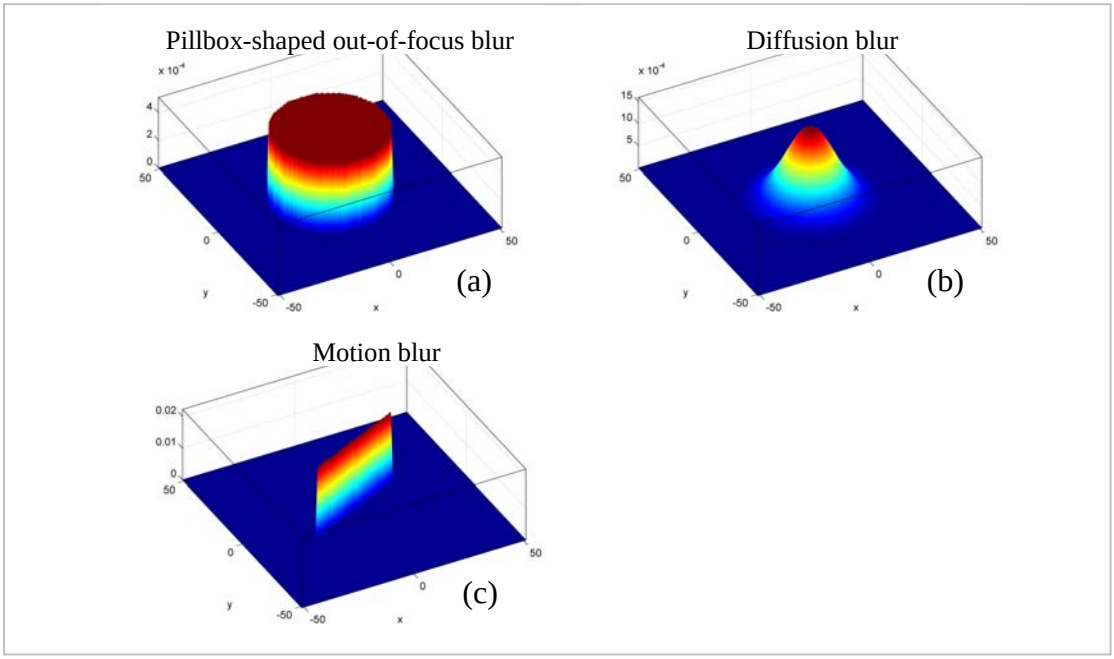


Figure 2-4 Examples of impulse responses of (a) pillbox-shaped out-of-focus blur, (b) diffusion (Gaussian) blur and (c) motion (directional) blur at 25 degrees.

Example of impulse response of pillbox-shaped out-of-focus blur in (2.33), impulse response of diffusion blur in (2.34), and impulse response of motion blur in (2.35) are depicted, respectively, in (a), (b) and (c) of Figure 2-4.

2.6.3 PSF in SAR autofocus

The point spread function in SAR [28, 29] is naturally defined in the spatial frequency (angular wavenumber) domain or so called the k-space. The k-space for two-dimensional SAR imaging has two orthogonal axes, say k_r and k_c, which are frequency variables for range and cross-range respectively. The PSF for the SAR autofocus problem [31, 32, 36-40] in the spatial frequency domain is expressed as

$$H(k_r, k_c) = e^{j\phi(k_c)}. \tag{2.36}$$

Note that the point spread function in SAR autofocus is generally phase only and in the cross-range dimension. However, for inverse synthetic aperture radar (ISAR) autofocus, a more general model is usually required due to non-uniform target rotation.

2.6.4 PSF in ISAR autofocus

In Section 2.5.3, we described the ISAR autofocus problem. This interpretation suggests that in order to capture both the azimuth and range(phase) error the point spread function can be modeled by a two-dimensional complex impulse response in the spatial domain, $h(r,c)$, with finite support. In the spatial frequency domain, this PSF can be expressed as

$$H(k_r, k_c) = A(k_r, k_c)\, e^{i\phi(k_r, k_c)} \tag{2.37}$$

where $A(k_r, k_c)$ is the amplitude and $\phi(k_r, k_c)$ is the phase component of the PSF in the spatial frequency domain.

This error model is crucial for our proposed ISAR autofocus approach. It justifies our belief that the Bussgang deconvolution framework is fundamentally applicable to the ISAR autofocus problem.

3 Self-correcting multi-channel Bussgang (SCMB) blind deconvolution

In this chapter, we introduce the idea of extending the Bussgang blind deconvolution framework to applications that do not meet both the independent and identically distributed (iid) and known probability density function (pdf) requirements on the input signal. We start by investigating the possibility of relaxing the iid requirement and then follow by modifying the original Bussgang algorithm by adding a feedback loop and a pdf parameter estimator so as to remove the known pdf requirement. The input signal is modeled using a Gaussian mixture model; the expectation maximization (EM) algorithm is used for the parameters estimation. In addition, a link between the multi-channel Bussgang algorithm and the longer equivalent equalizer length is established which suggests that the multi-channel Bussgang blind deconvolution algorithm converges more reliably to the global optimum [2] than its single-channel counterpart. We call the resulting design the self-correcting multi-channel Bussgang (SCMB) blind deconvolution algorithm. Its functional block diagram is shown in Figure 3-1 below.

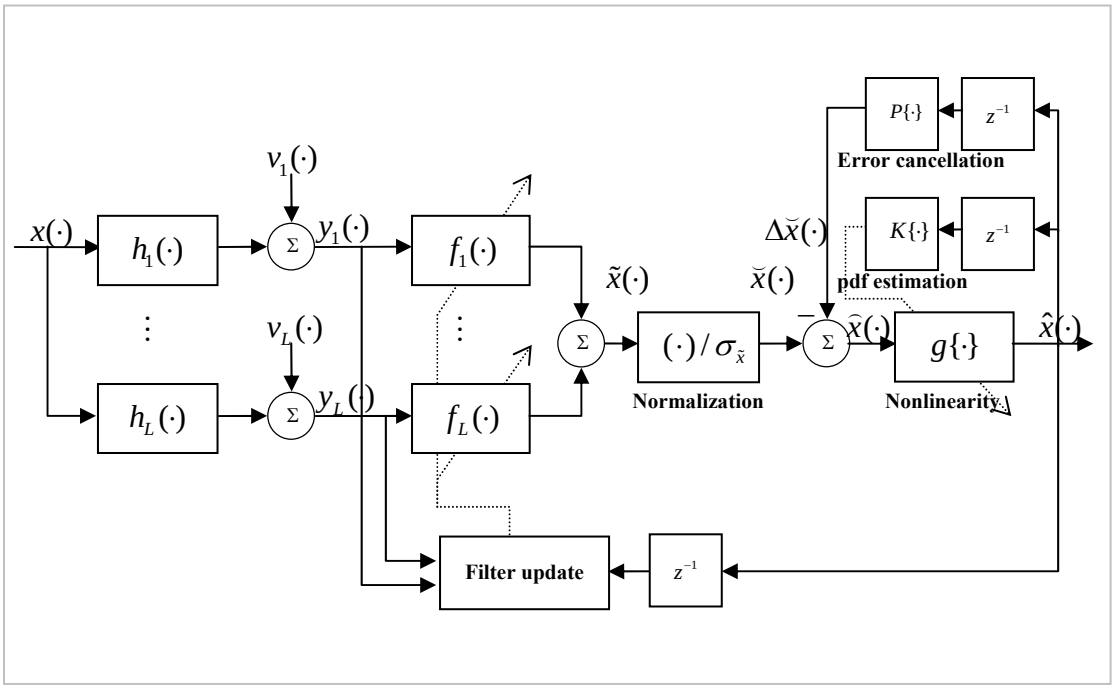


Figure 3-1 Functional block diagram of the self-correcting multi-channel Bussgang (SCMB) blind deconvolution algorithm

The variables in the functional block diagram of the self-correcting multi-channel Bussgang (SCMB) blind deconvolution algorithm in Figure 3-1 are as follow:

$x(\cdot)$	- Unknown input data.
$h_i(\cdot)$	- Unknown impulse response or point spread function (PSF) at channel i.
$v_i(\cdot)$	- Additive noise at channel i.
$y_i(\cdot)$	- Measurable output data at channel i.
$f_i(\cdot)$	- Inverse impulse response or equalization filter at channel i.
$\tilde{x}(\cdot)$	- Deconvolved output before normalization.
$\breve{x}(\cdot)$	- Deconvolved output with unity variance.
$\Delta\breve{x}(\cdot)$	- Feedback signal for pdf error cancellation.
$g\{\cdot\}$	- Zero-memory nonlinearity.
$K\{\cdot\}$	- Process for estimation of pdf parameters using EM algorithm.
$P\{\cdot\}$	- Feedback mechanism for pdf parameters error cancellation.
$\hat{x}(\cdot)$	- Output of the Bussgang algorithm (estimation of $x(n)$) (the desired signal in the context of adaptive filtering).
z^{-1}	- Unit frame (iteration) delay.

Note that in the SCMB blind deconvolution algorithm, as shown in the above functional block diagram, the error cancellation feedback loop, the pdf estimation and, of course, the Bussgang nonlinearity will vary with applications. More application-specific details of these modules will be discussed in later chapters. In Chapter 4, details specific to the binary image restoration problem will be discussed, and in Chapter 5, details specific to the SAR/ISAR autofocus problem will be discussed.

3.1 Relaxing the iid requirement for Bussgang blind deconvolution

The independent and identically distributed (iid) requirement of the input data, $x(n)$, is imposed in implementations of the Bussgang algorithm discussed in the literature [1, 12, 41], so that the deconvolution noise, $w(n) = \breve{x}(n) - x(n)$, will be zero mean, white and Gaussian. We suggest here that while the iid requirement on the input data is a sufficient condition, it is not a necessary one. Therefore, it can be relaxed. We illustrate our assertion for the single channel case with no additive noise, as shown in Figure 3-2, for simplicity.

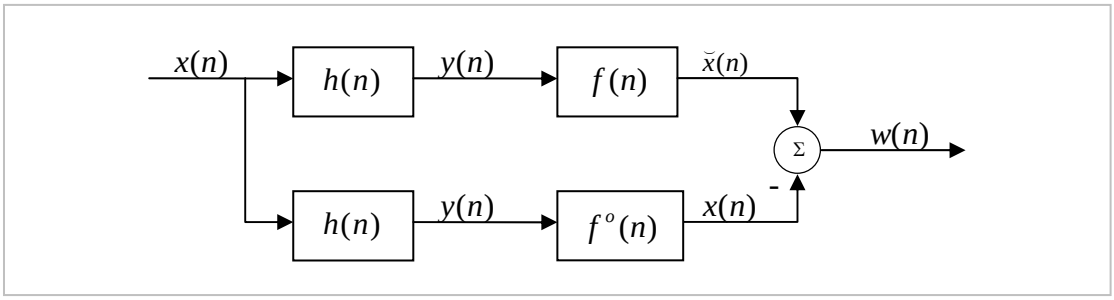


Figure 3-2 Functional block diagram for generating the deconvolution noise for the single channel case with no additive noise

Note that $f^o(n)$ is the ideal inverse impulse response such that $h(n) * f^o(n) = \delta(n)$, where $*$ represents convolution and $\delta(n)$ represents the unit impulse function with $\delta(0) = 1$ and $\delta(n) = 0$ for all $n \neq 0$. We can then express the deconvolution noise as

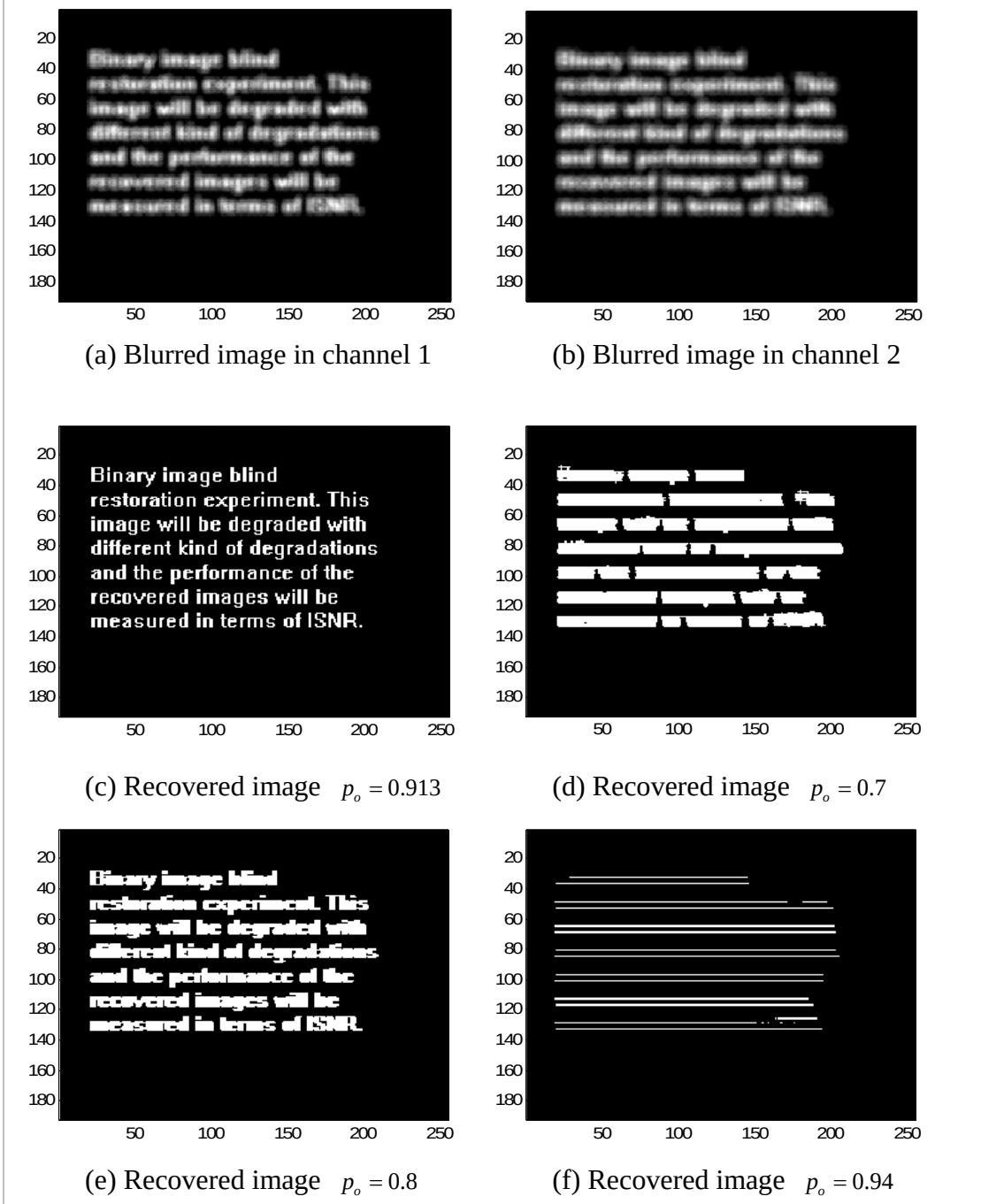
$$w(n) = \sum_{\substack{l=k+i \\ k\in[0,L_f-1] \\ i\in[0,L_h-1]}} \nabla(l)\, x(n-l), \tag{3.1}$$

where $\nabla(l) = [f(k) - f^o(k)]h(i)$ is a noise-like equalization residue, L_f is the filter length of the inverse impulse response and L_h is the filter length of the impulse response. For sufficiently long filter lengths L_h and L_f, when close to convergence, we propose to approximate the noise-like equalization residue $\nabla(l)$ by a zero-mean white Gaussian distribution. This is a weaker assumption than the iid assumption on the input data $x(n)$ that is found in the literature. Instead, we view the input $x(n)$ as deterministic. Therefore, from (3.1) the deconvolution noise $w(n)$ is a linear combination of zero-mean white Gaussian random variables $\nabla(l)$ which means that it is zero mean, white, and Gaussian. As a result, Bussgang blind deconvolution should be applicable to a non-white, correlated input signal.

3.2 Dealing with the unknown pdf in Bussgang blind deconvolution

In digital communication, the pdf of the transmitted signal can quite often be adequately modeled by a uniform distribution, because the transmitted symbols are independent and identically distributed (iid). In this case, the pdf can be completely known. However, for applications other than digital communication, such as image restoration, the pdf of the input data won't be completely known and will vary from application to application. To

demonstrate the problem of an unknown pdf in Bussgang blind deoconvolution, we apply the algorithm to a binary image restoration problem for which the pdf of the original binary image is not completely known. Even though the pdf of the binary image can be expressed in the form $p_x(x) = p_0\delta(x) + (1-p_0)\delta(x-a)$ with the pixel value of either 0 or a, the probability of the background p_o will not be known, since the original, undistorted image will not be available. In this example, the original binary image has a background probability of $p_o = 0.913$ and is blurred by a 7x7 and a 9x9 out-of-focus point spread function (PSF) as defined in (2.33), respectively, in channels one and two, as shown in Figure 3-3(a) and Figure 3-3(b). The multi-channel Bussgang algorithm is applied with different guesses for p_o and the restored images are shown in Figure 3-3 (c-f). In Figure 3-3 (c), we show that the multi-channel Bussgang blind deconvolution algorithm restores the original image flawlessly as in the literature [42] when the correct value of p_o is known. However, this is actually not very meaningful or useful, since, we will not know what the background probability p_o should be in reality. Guesses with deviations of 0.213, 0.113 or even as little as 0.027 from $p_o = 0.913$ give bad restoration results as shown in Figure 3-3 (d-f), respectively. As a result, we conclude that Bussgang blind deconvolution cannot be applied to binary image restoration without modification.



(a) Blurred image in channel 1

(b) Blurred image in channel 2

(c) Recovered image $p_o = 0.913$

(d) Recovered image $p_o = 0.7$

(e) Recovered image $p_o = 0.8$

(f) Recovered image $p_o = 0.94$

Figure 3-3 Problem of unknown pdf in Bussgang blind deconvolution

3.3 Design of a feedback mechanism for the Bussgang algorithm

In Section 3.2, we illustrated the problem of the multi-channel blind deconvolution algorithm, when total knowledge of the pdf of the input signal is not available, using a binary image restoration example. In fact, most blind deconvolution problems do not have complete knowledge of the pdf of the input signal. Therefore, despite the many theoretical advantages of the Bussgang blind deconvolution algorithm, its popularity is very limited. To change this picture, we attempt here to remove the known pdf limitation by designing a feedback loop to cancel out, or self-correct, the output error caused by the incomplete knowledge of the pdf. The intent of this feedback mechanism is to estimate the pdf simultaneously and harmoniously with the estimation of the equalization filters in the original Bussgang algorithm without jeopardizing its convergence. To explain the concept more precisely, let the pdf of the input signal be paramterized by a vector Φ. The Bussgang nonlinearity can then be expressed as,

$$\hat{x}(\cdot) = g\left\{\breve{x}(\cdot);\Phi\right\}. \tag{3.2}$$

An estimation error in Φ at iteration i, $\Delta\Phi^{(i)}$, will result in a nonlinearity output error $\Delta\hat{x}(\cdot)^{(i)}$. That is,

$$\hat{x}(\cdot) + \Delta x(\cdot)^{(i)} = g\left\{\breve{x}(\cdot);\Phi + \Delta\Phi^{(i)}\right\}. \tag{3.3}$$

Therefore, we introduce the negative feedback term, $\Delta\breve{x}(\cdot)^{(i)}$, in an attempt to cancel out the nonlinearity output error such that,

$$\hat{x}(\cdot) = g\left\{\breve{x}(\cdot) - \Delta\breve{x}(\cdot)^{(i)};\Phi + \Delta\Phi^{(i)}\right\}. \tag{3.4}$$

The question now is how to find the negative feedback signal $\Delta\breve{x}(\cdot)^{(i)}$. This unconventional feedback loop design is application dependent. A more detailed discussion of the design for the binary image restoration problem and for the SAR/ISAR autofocus problem are postponed to Chapters 4 and 5, respectively.

3.4 Gaussian mixture pdf modeling and estimation using the EM algorithm

In the SCMB blind deconvolution algorithm, a Gaussian mixture model is used to parameterize either a portion of or the whole pdf of the input signal depending on the application. For example, in the binary image restoration problem, the binary image is modeled by a two-component Gaussian mixture model, while in the SAR/ISAR autofocus problem, the logarithmic amplitude of the complex SAR/ISAR image is modeled by a three-component Gaussian mixture model. Therefore, estimating the unknown pdf is reduced to estimating the parameters of the Gaussian mixture model. The general formula for the pdf of a Gaussian mixture model of random variable z is

$$p_z(z) = \sum_{c=1}^{C} w_c \left[\left(\frac{1}{\sqrt{2\pi}\,\sigma_c} e^{-\frac{(z-\mu_c)^2}{2\sigma_c^2}} \right) \right], \tag{3.5}$$

where

$$\sum_{c=1}^{C} w_c = 1. \tag{3.6}$$

The number of Gaussian components, C, depends on the application. The parameters of the Gaussian mixture model are the means, standard deviations and weights respectively μ_c, σ_c and w_c. They are estimated utilizing the expectation maximization (EM) algorithm as described in Section 2.2.3.

3.5 Global convergence for the multi-channel case

The global convergence of the original Bussgang blind deconvolution algorithm was proved by Albert Benveniste in 1980 [2] for an infinite length equalization filter. The multi-channel Bussgang algorithm effectively increases the equalization filter length compared to its single-channel counterpart and, therefore, can be expected to converge better. The approximation of the deconvolution noise $w(\cdot)$ by a zero-mean, white Gaussian distribution is also expected to be more appropriate in the multi-channel case, since multiple noise-like equalization residues, $\nabla_c(l)$, are summed as compared to the single-channel case in (3.1). In the one-dimensional multi-channel case, the expression for the deconvolution noise becomes

$$\sum_{\substack{l=k+i \\ k\in[0,L_f-1] \\ i\in[0,L_h-1]}} \left[\sum_{c=1}^{C} \nabla_c(l) \right] x(n-l)\,, \tag{3.7}$$

where the equalization residue for each channel is

$$\nabla_c(l) = \left[f_c(k) - f_c^{o}(k) \right] h_c(i)\,. \tag{3.8}$$

3.6 Bussgang nonlinearity in SCMB algorithm

In the derivation of Bussgang blind deconvolution algorithm, the nonlinearity $g\{\cdot\}$ results from Bayesian estimation, in which the Bayes risk $J(\hat{\mathbf{x}})$, defined in (2.12), is minimized. The nonlinearity depends on the choice of the cost functional $C(\cdot)$ in the Bayes risk.

3.6.1 Bayesian estimation with quadratic cost

In our self-correcting multi-channel Bussgang (SCMB) blind deconvolution algorithm, we choose a quadratic cost functional for the Bayes risk. That is,

$$C(x - \hat{x}(\breve{x})) = (x - \hat{x}(\breve{x}))^2\,, \tag{3.9}$$

where, x, $\breve{x}$ and $\hat{x}$ are the random variables for the unknown original signal (input signal), the output of the equalization filters and the output of the Bussgang nonlinearity, respectively. With the choice of a quadratic cost functional, the Bayesian estimate is also the minimum mean square error (MMSE) estimation. By using variational calculus and substituting (3.9) into (2.12), we have

$$\hat{x}(\breve{x}) = E\{x \mid \breve{x}\} = \int_{-\infty}^{\infty} x\, p_{x|\breve{x}}(x \mid \breve{x}) dx \triangleq g\{\breve{x}\}\,, \tag{3.10}$$

where $p_{x|\breve{x}}(x \mid \breve{x})$ is the conditional probability density function of x when $\breve{x}$ is known. Therefore, the output of the SCMB blind deconvolution algorithm, which is the Bussgang nonlinearity, $g\{\breve{x}\}$, is the conditional mean of the inaccessible input signal, x, under the condition that the equalization filter output $\breve{x}$ is known; its derivation is given in Appendix **Error! Reference source not found.**.

3.6.2 Expressing the nonlinearity in terms of the pdf

The nonlinearity $g\{\breve{x}\} \triangleq E\{x \mid \breve{x}\}$ in the previous section is more usefully expressed in terms of the pdf of the input signal x and the pdf of the deconvolution noise, w. By using Bayes' rule, the definition of the deconvolution noise $w = \breve{x} - x$, and the relationship of the probability density functions $p_{\breve{x}}(\breve{x}) = p_w(\breve{x}) * p_x(\breve{x})$ caused by the sum of independent random variables $\breve{x} = w + x$, it can be shown that

$$g\{\breve{x}\} \triangleq E\{x \mid \breve{x}\} = \frac{\int x\, p_x(x)\, p_w(\breve{x} - x)\, dx}{\int p_x(x)\, p_w(\breve{x} - x)\, dx}. \tag{3.11}$$

The derivation of the above fundamental Bussgang nonlinearity equation is given in Appendix **Error! Reference source not found.**.

3.7 Equalization filter updating in the SCMB algorithm

The equalization filter update in the Bussgang algorithm usually employs a stochastic gradient-based adaptive filtering technique. A stream-based digital signal processing (DSP) technique such as the least mean square (LMS) algorithm is often used. The choice between stream-based versus frame-based approaches for updating the equalization filter is analogous to the direct convolution versus an FFT implementation of an FIR filter. The price that frame-based approaches impose is that the output is delayed by a frame interval. However, with the self-correcting multi-channel Bussgang (SCMB) algorithm, a frame-based approach is more suitable, since the feedback mechanism already requires a frame delay to exploit the statistics collected from the previous frame. Also for frame-based applications, such as image or video processing, a frame-based approach fits the problem naturally.

3.7.1 Estimated deconvolution noise

To develop an update rule for the equalization filter $f_c(\cdot)$, it is reasonable to seek to minimize the energy of the deconvolution noise $w(\cdot) = \breve{x}(\cdot) - x(\cdot)$. However, the deconvolution noise is not computable since the input signal $x(\cdot)$ is always unknown. Instead we minimize the energy of the so- called estimated deconvolution noise, $\varepsilon(\cdot)$, that we define as

$$\varepsilon(\cdot) = \breve{x}(\cdot) - \hat{x}(\cdot), \tag{3.12}$$

which is the difference between the normalized deconvolution output and the nonlinearity output.

By using variational calculus, the mean square error (MSE) $E\{\varepsilon(\cdot)\varepsilon(\cdot)^*\}$ of the estimated deconvolution noise is minimized with respect to the equalization filter coefficients under the assumption that the equalization filters are independent of the estimated deconvolution noise. We conclude that the estimated deconvolution noise is orthogonal to the input of the equalization filter. That is, in the two-dimensional case,

$$E\{y_c(m-k, n-l)\varepsilon(m,n)^*\} = 0, \qquad \forall c \tag{3.13}$$

where $y_c(m,n)$ is the input of the equalization filter $f_c(m,n)$ at the c-th channel.

3.7.2 Multi-channel Wiener filter

Beginning with the orthogonality principle in (3.13) and substituting for the estimated deconvolution noise $\varepsilon(m,n)$,

$$\varepsilon(m,n) = \sum_c \sum_{p,q} f_c(p,q) y_c(m-p, n-q) - \hat{x}(m,n), \tag{3.14}$$

then after some minor manipulation of the equation the equalization filter coefficients that minimize the estimated equalization error can be seen to satisfy the equations

$$\sum_{c=1}^{C} \left[\sum_{p=-P}^{P} \sum_{q=-Q}^{Q} f_c(p,q) r_{y_c y_i}(m-p, n-q) \right] = r_{\hat{x} y_i}(m,n), \tag{3.15}$$

where the channel index $i \in [1, C]$. The equalization filter coefficient indexes m and n extend over the range $m \in [-P, P]$ and $n \in [-Q, Q]$, and the two-dimensional cross-correlations are given by

$$r_{y_c y_i}(p,q) = E\{y_c(m,n) y_i(m-p, n-q)^*\}, \tag{3.16}$$

and

$$r_{\hat{x} y_i}(p,q) = E\{\hat{x}(m,n) y_i(m-p, n-q)^*\}. \tag{3.17}$$

It is clear that solving for the equalization filter f_c is equivalent to finding the solution for the multi-channel Wiener-Hopf equation in (3.15). In the actual implementation, equation (3.15) is expressed in a compact matrix form as shown below,

$$\mathbf{R}_y \vec{f} = \vec{R}_{\hat{x}y}, \tag{3.18}$$

where the dimension of $\mathbf{R}_y$ is $C(2P+1)(2Q+1)\times C(2P+1)(2Q+1)$ and the dimensions of $\vec{f}$ and $\vec{R}_{\hat{x}y}$ are both $C(2P+1)(2Q+1)\times 1$.

The multi-channel equalization filter vector $\vec{f}$ is formed by stacking the column scan of the two-dimensional equalization filter $\mathbf{f}_j$ from each channel as described below,

$$\mathbf{f}_j = \begin{bmatrix} f_j(-P,-Q) & \cdots & f_j(-P,Q) \\ \vdots & \ddots & \vdots \\ f_j(P,-Q) & \cdots & f_j(P,Q) \end{bmatrix}, \quad \vec{f}_j = Vec\{\mathbf{f}_j\} \text{ , and } \quad \vec{f} = \begin{bmatrix} \vec{f}_1 \\ \vdots \\ \vec{f}_C \end{bmatrix} \tag{3.19}$$

where $Vec\{\cdot\}$ represents the column scan of a matrix.

The multi-channel input correlation matrix $\mathbf{R}_y$ used in the compact matrix form of the multi-channel Wiener-Hopf equation in (3.18) is composed of a hierarchy of matrices as shown below,

$$\tilde{\mathbf{R}}_{ij}(m-n) = \begin{bmatrix} r_{y_j y_i}(0,m-n) & r_{y_j y_i}(-1,m-n) & \cdots & r_{y_j y_i}(-2P,m-n) \\ r_{y_j y_i}(1,m-n) & r_{y_j y_i}(0,m-n) & & r_{y_j y_i}(-2P+1,m-n) \\ \vdots & & \ddots & \vdots \\ r_{y_j y_i}(2P,m-n) & r_{y_j y_i}(2P-1,m-n) & \cdots & r_{y_j y_i}(0,m-n) \end{bmatrix} \tag{3.20}$$

$$\mathbf{R}_{ij} = \begin{bmatrix} \tilde{\mathbf{R}}_{ij}(0) & \tilde{\mathbf{R}}_{ij}(-1) & \cdots & \tilde{\mathbf{R}}_{ij}(-2Q) \\ \tilde{\mathbf{R}}_{ij}(1) & \tilde{\mathbf{R}}_{ij}(0) & & \tilde{\mathbf{R}}_{ij}(-2Q+1) \\ \vdots & & \ddots & \vdots \\ \tilde{\mathbf{R}}_{ij}(2Q) & \tilde{\mathbf{R}}_{ij}(2Q-1) & \cdots & \tilde{\mathbf{R}}_{ij}(0) \end{bmatrix} \text{ and} \tag{3.21}$$

$$\mathbf{R}_y = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \cdots & \mathbf{R}_{1C} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & & \mathbf{R}_{2C} \\ \vdots & & \ddots & \vdots \\ \mathbf{R}_{C1} & \mathbf{R}_{C2} & \cdots & \mathbf{R}_{CC} \end{bmatrix}. \tag{3.22}$$

$\tilde{\mathbf{R}}_{ij}(m,n)$ is a Toeplitz matrix, $\mathbf{R}_{ij}$ is a block Toeplitz matrix and $\mathbf{R}_y$ is a tiled block Toeplitz matrix.

Similar to the procedure used to construct the multi-channel equalization filter vector $\vec{f}$, the multi-channel cross-correlation vector between the output and the input, $\vec{R}_{\hat{x}y}$, is formed by stacking the column scan of the two-dimensional cross-correlation $\mathbf{R}_{\hat{x}y_c}$ from each channel as shown below,

$$\mathbf{R}_{\hat{x}y_c} = \begin{bmatrix} r_{\hat{x}y_c}(-P,-Q) & \cdots & r_{xy_c}(-P,Q) \\ \vdots & \ddots & \vdots \\ r_{\hat{x}y_c}(P,-Q) & \cdots & r_{xy_c}(P,Q) \end{bmatrix}, \quad \vec{R}_{xy_c} = Vec\{\mathbf{R}_{xy_c}\} \quad \text{and} \quad \vec{R}_{\hat{x}y} = \begin{bmatrix} \vec{R}_{\hat{x}y_1} \\ \vdots \\ \vec{R}_{\hat{x}y_C} \end{bmatrix}. \quad (3.23)$$

By solving the compact matrix form of the Wiener-Hopf equation using direct matrix inverse of $\mathbf{R}_y$ in (3.18), the multi-channel equalization filter update for the SCMB algorithm can be accomplished.

3.7.3 Efficient multi-channel Wiener filter solver utilizing the FFT

The update of the single-channel or the multi-channel equalization filter of the proposed SCMB algorithm requires solving the Wiener-Hopf equation. When the support of the two-dimensional equalization filter is large, especially in the multi-channel case, solving for the equalization filter using (3.18) directly, which does not exploit the structure of the input correlation matrix $\mathbf{R}_y$, can be exceptionally costly in terms of both the memory and the computation time required. For example, in some cases, it is desirable to have a two-dimensional equalization filter with the support size as large as the image, say 512×512. Assuming a two-channel scenario, the matrix $\mathbf{R}_y$ that needs to be inverted in (3.18) would have a dimension of 524288×524288. Thus, it is obviously worthwhile to pursue a more memory efficient approach to solving for the equalization filter.

In the single-channel case, the multi-channel input correlation matrix, $\mathbf{R}_y$ in (3.18), reduces to $\mathbf{R}_{11}$ as shown in (3.22), which is a block Toeplitz matrix. Fast algorithms exist in the literature that take advantage of the block Toeplitz structure [43-46]. Most of these fast algorithms are deterministic, iterative methods that make use of both the matrix inverse lemma and the fact that both the block Toeplitz matrix, $\mathbf{R}_{11}$, and its inverse, $\mathbf{R}_{11}^{-1}$, are block persymmetric matrices. That is,

$$J_{(2P+1),(2Q+1)}\left(\mathbf{R}_{11}\right)^T J_{(2P+1),(2Q+1)} = \left(\mathbf{R}_{11}\right), \quad (3.24)$$

and

$$J_{(2P+1),(2Q+1)}\left(\mathbf{R}_{11}^{-1}\right)^{T} J_{(2P+1),(2Q+1)} = \left(\mathbf{R}_{11}^{-1}\right), \tag{3.25}$$

where $J_{(2P+1),(2Q+1)}$ is the block exchange matrix as shown below,

$$J_{(2P+1),(2Q+1)} = \underbrace{\begin{bmatrix} 0 & \cdots & 0 & J_{(2P+1)} \\ \vdots & & J_{(2P+1)} & 0 \\ 0 & \iddots & & \vdots \\ J_{(2P+1)} & 0 & \cdots & 0 \end{bmatrix}}_{(2Q+1)}, \quad J_{(2P+1)} = \underbrace{\begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & & 1 & 0 \\ 0 & \iddots & & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix}}_{(2P+1)}. \tag{3.26}$$

These fast algorithms can be generalized into the Trench algorithm and the reverse Trench algorithm. In the Trench algorithm approach, the column scan of the two-dimensional equalization filter $\mathbf{f}_1$, $\vec{f}_1$, of dimension $(2P+1)(2Q+1)\times 1$, starts with the dimension of a single column of length $(2P+1)$ and grows by adding $(2P+1)$ elements to its length with each iteration for $2Q$ iterations. In contrast, for the reverse Trench algorithm approach, the algorithm first patches the block Toeplitz matrix $\mathbf{R}_{11}$ into a block circulant matrix, $\breve{\mathbf{R}}_{11}$, with a dimension of $(2P+1)\ (4Q+1) \times (2P+1)\ (4Q+1)$. Because of the block circulant structure, its inverse $\breve{\mathbf{Q}}_{11}$ is also a block circulant matrix, which can be efficiently obtained by using the Fast Fourier Transform (FFT). The dimension of $\breve{\mathbf{Q}}_{11}$ then shrinks by a dimension of $(2P+1)$ elements with each iteration for $2Q$ iterations before reaching the final solution.

In the single channel implementation of the equalization filter update for the SCMB algorithm, both the Trench algorithm and the reverse Trench algorithm-based approaches are utilized to reduce the required memory and computation time. We will first outline the Trench algorithm-based approach here and will strip off the subscript representing the channel number that we use in the general multi-channel case. Then, equations (3.21) and (3.22) become

$$\mathbf{R}_{2Q+1} = \begin{bmatrix} \tilde{\mathbf{R}}(0) & \tilde{\mathbf{R}}(-1) & \cdots & \tilde{\mathbf{R}}(-2Q+1) & \tilde{\mathbf{R}}(-2Q) \\ \tilde{\mathbf{R}}(1) & \tilde{\mathbf{R}}(0) & & \tilde{\mathbf{R}}(-2Q+2) & \tilde{\mathbf{R}}(-2Q+1) \\ \vdots & & \ddots & & \vdots \\ \tilde{\mathbf{R}}(2Q-1) & \tilde{\mathbf{R}}(2Q-2) & & \tilde{\mathbf{R}}(0) & \tilde{\mathbf{R}}(-1) \\ \tilde{\mathbf{R}}(2Q) & \tilde{\mathbf{R}}(2Q-1) & \cdots & \tilde{\mathbf{R}}(1) & \tilde{\mathbf{R}}(0) \end{bmatrix}. \tag{3.27}$$

Noticing that $\mathbf{R}_y$ is replaced by $\mathbf{R}_{2Q+1}$, we replace the subscript y with the number of rows or columns of the Toeplitz matrices in $\mathbf{R}_y$ to facilitate a nested structure representation as shown below:

$$\mathbf{R}_{2Q+1} = \begin{bmatrix} \mathbf{R}_{2Q} & \underline{\tilde{\mathbf{R}}}_{-2Q} \\ \underline{\tilde{\mathbf{R}}}_{2Q}{}^{T} & \tilde{\mathbf{R}}(0) \end{bmatrix}, \tag{3.28}$$

where

$$\mathbf{R}_{2Q} = \begin{bmatrix} \tilde{\mathbf{R}}(0) & \tilde{\mathbf{R}}(-1) & \cdots & \tilde{\mathbf{R}}(-2Q+1) \\ \tilde{\mathbf{R}}(1) & \tilde{\mathbf{R}}(0) & & \tilde{\mathbf{R}}(-2Q+2) \\ \vdots & & \ddots & \\ \tilde{\mathbf{R}}(2Q-1) & \tilde{\mathbf{R}}(2Q-2) & & \tilde{\mathbf{R}}(0) \end{bmatrix}, \tag{3.29}$$

and

$$\underline{\tilde{\mathbf{R}}}_{-2Q} = \begin{bmatrix} \tilde{\mathbf{R}}(-2Q) \\ \tilde{\mathbf{R}}(-2Q+1) \\ \vdots \\ \tilde{\mathbf{R}}(-1) \end{bmatrix}, \quad \underline{\tilde{\mathbf{R}}}_{2Q} = \begin{bmatrix} \tilde{\mathbf{R}}(2Q)^T \\ \tilde{\mathbf{R}}(2Q-1)^T \\ \vdots \\ \tilde{\mathbf{R}}(1)^T \end{bmatrix}. \tag{3.30}$$

The Wiener-Hopf equation in (3.18) can be expressed in the single channel case with the channel number dropped,

$$\mathbf{R}_{2Q+1}\,\vec{f} = \vec{R}_{\hat{x}y} \tag{3.31}$$

where,

$$\vec{f} = \begin{bmatrix} \mathbf{f}(1) \\ \mathbf{f}(2) \\ \vdots \\ \mathbf{f}(2Q+1) \end{bmatrix}, \textit{ and } \quad \vec{R}_{\hat{x}y} = \begin{bmatrix} \mathbf{R}_{\hat{x}y}(1) \\ \mathbf{R}_{\hat{x}y}(2) \\ \vdots \\ \mathbf{R}_{\hat{x}y}(2Q+1) \end{bmatrix}. \tag{3.32}$$

Note that $\mathbf{f}$ and $\mathbf{R}_{\hat{x}y}$ are, respectively, the two-dimensional equalization filter and the cross-correlation matrix, as defined in (3.19) and (3.23), with the channel subscript dropped. $\mathbf{f}(n)$ and $\mathbf{R}_{\hat{x}y}(n)$ are the $n-th$ column of the matrix $\mathbf{f}$ and $\mathbf{R}_{\hat{x}y}$, respectively. $\underline{\mathbf{f}}(n)$ and $\underline{\mathbf{R}}_{\hat{x}y}(n)$ are, respectively, the vectorized forms of $\mathbf{f}$ and $\mathbf{R}_{\hat{x}y}$, such that,

$$\underline{\mathbf{f}}(n)=\begin{bmatrix}\mathbf{f}(1)\\ \mathbf{f}(2)\\ \vdots\\ \mathbf{f}(n)\end{bmatrix},\ \textit{and}\quad \underline{\mathbf{R}}_{\hat{x}y}(n)=\begin{bmatrix}\mathbf{R}_{\hat{x}y}(1)\\ \mathbf{R}_{\hat{x}y}(2)\\ \vdots\\ \mathbf{R}_{\hat{x}y}(n)\end{bmatrix}. \tag{3.33}$$

Equations (3.28), (3.29) and (3.30) can be rewritten in their iterative form at the $n-th$ iteration, respectively as,

$$\mathbf{R}_{n+1}=\begin{bmatrix}\mathbf{R}_n & \tilde{\underline{\mathbf{R}}}_{-n}\\ \tilde{\underline{\mathbf{R}}}_n{}^T & \tilde{\mathbf{R}}(0)\end{bmatrix} \tag{3.34}$$

$$\mathbf{R}_n=\begin{bmatrix}\tilde{\mathbf{R}}(0) & \tilde{\mathbf{R}}(-1) & \cdots & \tilde{\mathbf{R}}(-n+1)\\ \tilde{\mathbf{R}}(1) & \tilde{\mathbf{R}}(0) & & \tilde{\mathbf{R}}(-n+2)\\ \vdots & & \ddots & \\ \tilde{\mathbf{R}}(n-1) & \tilde{\mathbf{R}}(n-2) & & \tilde{\mathbf{R}}(0)\end{bmatrix} \tag{3.35}$$

$$\tilde{\underline{\mathbf{R}}}_{-n}=\begin{bmatrix}\tilde{\mathbf{R}}(-n)\\ \tilde{\mathbf{R}}(-n+1)\\ \vdots\\ \tilde{\mathbf{R}}(-1)\end{bmatrix},\quad \tilde{\underline{\mathbf{R}}}_n=\begin{bmatrix}\tilde{\mathbf{R}}(n)^T\\ \tilde{\mathbf{R}}(n-1)^T\\ \vdots\\ \tilde{\mathbf{R}}(1)^T\end{bmatrix} \tag{3.36}$$

With the variables needed defined above, the Trench algorithm-based approach [46] used to solve the Wiener-Hopf equation in (3.31) for the equalization filter in the SCMB algorithm is given in the table below:

Initialization		
	$\underline{w}_1=-\tilde{\mathbf{R}}(0)\,\tilde{\mathbf{R}}(-1)$	
	$\underline{v}_1=-\tilde{\mathbf{R}}(0)^{-T}\,\tilde{\mathbf{R}}(1)^T$	
	$\alpha_1=\tilde{\mathbf{R}}(0)+\tilde{\mathbf{R}}(1)^T\,\underline{w}_1$	
	$\underline{\mathbf{f}}(1)=\tilde{\mathbf{R}}(0)^{-1}\,\underline{\mathbf{R}}_{\hat{x}y}(1)$	
Main iteration for $n=[1,\ldots,2Q]$		
	$\beta_n=\underline{v}_n^T\,\underline{\mathbf{R}}_{\hat{x}y}(n)+\mathbf{R}_{xy}(n+1)$	where,
	$\underline{\mathbf{f}}(n+1)=\begin{bmatrix}\underline{\mathbf{f}}(n)\\ 0\end{bmatrix}+\begin{bmatrix}\underline{w}(n)\\ I\end{bmatrix}\alpha_n^{-1}\beta_n$	$\hat{\tilde{\underline{\mathbf{R}}}}_{-n}=J_{(2P+1),(n)}\,\tilde{\underline{\mathbf{R}}}_{-n}$ $\hat{\tilde{\mathbf{R}}}(-(n+1))=J_{(2P+1)}\,\tilde{\mathbf{R}}(-(n+1))$ $\hat{I}=J_{(2P+1)}\,I$
	$\beta_{+n}=\underline{w}_n^T\,\hat{\tilde{\underline{\mathbf{R}}}}_{-n}+\tilde{\mathbf{R}}(-(n+1))$	

	$\underline{w}_{n+1} = \begin{bmatrix} 0 \\ \underline{w}_n \end{bmatrix} - \begin{bmatrix} \hat{I} \\ \underline{\hat{v}}_n \end{bmatrix} \alpha_n^{-T} \beta_{+n}$	
	$\beta_{-} = \underline{v}^T \tilde{\mathbf{R}}_{-} + \tilde{\mathbf{R}}(n+1)^T$	
	$\underline{\hat{v}}_{n+1} = \begin{bmatrix} \underline{\hat{v}}_n \\ 0 \end{bmatrix} - \begin{bmatrix} \underline{w}_n \\ I \end{bmatrix} \alpha_n^{-1} \beta_{-n}$	
	$\alpha_{n+1} = \alpha_n - \beta_{-n}^T \alpha_n^{-T} \beta_{+n}$	

Table 3-1 Iterative solution of a doubly block Toeplitz set of linear equations (Trench algorithm-based approach)

It turns out that there is a typographical error in one of the key equations in the original paper [46]. Instead of the step

$$\underline{w}_{n+1} = \begin{bmatrix} 0 \\ \underline{w}_n \end{bmatrix} - \begin{bmatrix} \hat{I} \\ \underline{\hat{v}}_n \end{bmatrix} \alpha_n^{-T} \beta_{+n} \tag{3.37}$$

as shown in the table above, the paper erroneously has,

$$\underline{w}_{n+1} = \begin{bmatrix} 0 \\ \underline{w}_n \end{bmatrix} - \begin{bmatrix} I \\ \underline{\hat{v}}_n \end{bmatrix} \alpha_n^{-T} \beta_{+n} \tag{3.38}$$

with $\hat{I} = J_{(2P+1)} I$ replaced by I. We determined this by re-deriving all the equations in [46] and comparing the differences.

In the multi-channel case, the multi-channel input correlation matrix $\mathbf{R}_y$ is a tiling of block Toeplitz matrices $\mathbf{R}_{ij}$ but itself is not a block Toeplitz matrix. As a result, both the Trench algorithm and the reverse Trench algorithm-based approaches are not applicable. Therefore, we developed an extremely efficient gradient-based method that can handle both the single channel and the multi-channel cases and does not require any matrix operations with a dimension of $C(2P+1)(2Q+1) \times C(2P+1)(2Q+1)$ as required by the direct method in (3.18). Instead, an FFT and an IFFT of dimension $(2(2P+1)-1) \times (2(2Q+1)-1)$ are used.

The outline of the derivation of our multi-channel gradient-based equalization filter update is given below. For simplicity, the support of the equalization filter is assumed to be $D_1 \times D_2$ instead of $(2P+1) \times (2Q+1)$, and the support of the input of the equalization filter is assumed to be $N_1 \times N_2$, such that $N_1 \geq D_1$ and $N_2 \geq D_2$, without loss of generosity. From the functional block diagram of the self-correcting multi-channel Bussgang (SCMB)

blind deconvolution algorithm in Figure 3-1, we define the error between the deconvolved output and the Bussgang nonlinearity output in the two-dimensional case as

$$e(n_1, n_2) = \hat{x}(n_1, n_2) - \tilde{x}(n_1, n_2)\,, \tag{3.39}$$

where $0 \le n_1 \le N_1 - 1$ and $0 \le n_2 \le N_2 - 1$, which can be more conveniently expressed in its vector form given by

$$\mathbf{e} = \hat{\mathbf{x}} - \tilde{\mathbf{x}}\,. \tag{3.40}$$

To update the equalization filter $\vec{f}_i^{\,(n)}$ of channel i at the $n-th$ iteration, the cost J, defined as the sum of the error energy samples, is minimized using a steepest descent approach,

$$\vec{f}_i^{\,(n+1)} = \vec{f}_i^{\,(n)} - \mu\left[\frac{\delta J}{\delta \vec{f}_i^{\,(n)}}\right], \tag{3.41}$$

where μ is the step size and

$$J = \mathbf{e}^H \mathbf{e}\,. \tag{3.42}$$

The deconvolved output, $\tilde{\mathbf{x}}$, can be written in terms of the input of the equalization filter, $y_i(n_1, n_2)$, and the equalization filter, $f_i(n_1, n_2)$, in its matrix-vector form as

$$\tilde{\mathbf{x}} = \sum_{i=1}^{C} \mathbf{Y}_i\, \vec{\mathbf{f}}_i\,. \tag{3.43}$$

$\mathbf{Y}_i$ is a $N_1 N_2 \times N_1 N_2$, block circulant matrix,

$$\mathbf{Y}_i = \underbrace{\begin{bmatrix} \tilde{\mathbf{Y}}_i(0) & \tilde{\mathbf{Y}}_i(N_2-1) & \cdots & \tilde{\mathbf{Y}}_i(1) \\ \tilde{\mathbf{Y}}_i(1) & \tilde{\mathbf{Y}}_i(0) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \tilde{\mathbf{Y}}_i(N_2-1) \\ \tilde{\mathbf{Y}}_i(N_2-1) & \cdots & \cdots & \tilde{\mathbf{Y}}_i(0) \end{bmatrix}}_{N_2 \text{ terms}}, \tag{3.44}$$

and $\tilde{\mathbf{Y}}_i(s)$ is a $N_1 \times N_1$ circulant matrix,

$$\tilde{\mathbf{Y}}_i(s) = \underbrace{\begin{bmatrix} y_i(0,s) & y_i(N_1-1,s) & \cdots & y_i(1,s) \\ y_i(1,s) & y_i(0,s) & \cdots & \vdots \\ \vdots & \vdots & \ddots & y_i(N_1-1,s) \\ y_i(N_1-1,s) & \cdots & \cdots & y_i(0,s) \end{bmatrix}}_{N_1 \text{ terms}}. \tag{3.45}$$

$\bar{\mathbf{f}}_i$ is a $N_1N_2 \times 1$ vector and is the column scan of the matrix $\bar{F}_i$ formed by zero-padding the equalization filter $f_i(n_1, n_2)$ from a support size of $D_1 \times D_2$ to a support size of $N_1 \times N_2$ which can be expressed as

$$\bar{\mathbf{f}}_i = vec\{\bar{\mathbf{F}}_i\}, \tag{3.46}$$

and

$$\bar{\mathbf{F}}_i = \underbrace{\begin{bmatrix} f_i(0,0) & \cdots & f_i(0,D_2-1) & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ f_i(D_1-1,0) & \cdots & f_i(D_1-1,D_2-1) & \cdots & 0 \\ \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}}_{N_1 \times N_2 \text{ matrix}}. \tag{3.47}$$

Now substituting (3.40) and (3.43) into (3.42), we have

$$J = \left(\hat{\mathbf{x}} - \sum_{i=1}^{C} \mathbf{Y}_i \bar{\mathbf{f}}_i \right)^H \left(\mathbf{x} - \sum_{j=1}^{C} \mathbf{Y}_j \bar{\mathbf{f}}_j \right). \tag{3.48}$$

By applying variational calculus on J, the gradient, $\frac{\delta J}{\delta \bar{\mathbf{f}}_i}$, becomes

$$\frac{\delta J}{\delta \bar{\mathbf{f}}_i} = -2\left[\mathbf{Y}_i^H \hat{\mathbf{x}} \right] + 2\left[\mathbf{Y}_i^H \mathbf{Y}_i \right] \bar{\mathbf{f}}_i + \sum_{j \neq i} \left[\mathbf{Y}_i^H \mathbf{Y}_j \right] \bar{\mathbf{f}}_j . \tag{3.49}$$

Notice that $\mathbf{Y}_i^H \mathbf{Y}_j$ from the expression above represents the two-dimensional circular cross-correlation between $y_i(n_1, n_2)$ and $y_j(n_1, n_2)$, which can be expressed as

$$\underbrace{\mathbf{Y}_i^H \mathbf{Y}_j}_{N_1N_2 \times N_1N_2} = BC\{\underbrace{\mathbf{R}_{y_i y_j}}_{N_1 \times N_2}\}, \tag{3.50}$$

where $BC\{\cdot\}$ is the block circulant operator. The cross-correlation matrix and its elements are respectively,

$$\mathbf{R}_{y_i y_j} = \underbrace{\begin{bmatrix} r_{y_i y_j}(-\left(\frac{N_1-1}{2}\right)^{\rceil}, -\left(\frac{N_2-1}{2}\right)^{\rceil}) & \cdots & r_{y_i y_j}(-\left(\frac{N_1-1}{2}\right)^{\rceil}, 0) & \cdots & r_{y_i y_j}(-\left(\frac{N_1-1}{2}\right)^{\rceil}, \left(\frac{N_2-1}{2}\right)^{\rceil}) \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ r_{y_i y_j}(0, -\left(\frac{N_2-1}{2}\right)^{\rceil}) & \cdots & r_{y_i y_j}(0,0) & \cdots & r_{y_i y_j}(0, \left(\frac{N_2-1}{2}\right)^{\rceil}) \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ r_{y_i y_j}(\left(\frac{N_1-1}{2}\right)^{\rceil}, -\left(\frac{N_2-1}{2}\right)^{\rceil}) & \cdots & r_{y_i y_j}(\left(\frac{N_1-1}{2}\right)^{\rceil}, 0) & \cdots & r_{y_i y_j}(\left(\frac{N_1-1}{2}\right)^{\rceil}, \left(\frac{N_2-1}{2}\right)^{\rceil}) \end{bmatrix}}_{N_1 \times N_2 \text{ matrix}}, \tag{3.51}$$

and

$$r_{y_i y_j}(r,s) = E\left\{ y_i(m,n)\, y_j(m-r,n-s) \right\}, \tag{3.52}$$

where $\lceil$ is the ceiling operator.

Similarly, $\mathbf{Y}_i^H \mathbf{Y}_i$ and $\mathbf{Y}_i^H \hat{\mathbf{x}}$ can be expressed as functions of the correlation matrices $\mathbf{R}_{y_i y_i}$ and $\mathbf{R}_{\hat{x} y_i}$ which are

$$\underbrace{\mathbf{Y}_i^H \mathbf{Y}_i}_{N_1 N_2 \times N_1 N_2} = BC\{\underbrace{\mathbf{R}_{y_i y_i}}_{N_1 \times N_2}\} \tag{3.53}$$

and

$$\mathbf{Y}_i^H \hat{\mathbf{x}} = vec\{\mathbf{R}_{\hat{x} y_i}\}. \tag{3.54}$$

Where the auto-correlation matrix $\mathbf{R}_{y_i y_i}$ and the cross-correlation function $\mathbf{R}_{\hat{x} y_i}$ are

$$\mathbf{R}_{y_i y_i} = \underbrace{\begin{bmatrix} r_{y_i y_i}(-\left(\frac{N_1-1}{2}\right)^{\lceil}, -\left(\frac{N_2-1}{2}\right)^{\lceil}) & \cdots & r_{y_i y_i}(-\left(\frac{N_1-1}{2}\right)^{\lceil}, 0) & \cdots & r_{y_i y_i}(-\left(\frac{N_1-1}{2}\right)^{\lceil}, \left(\frac{N_2-1}{2}\right)^{\lceil}) \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ r_{y_i y_i}(0, -\left(\frac{N_2-1}{2}\right)^{\lceil}) & \cdots & r_{y_i y_i}(0,0) & \cdots & r_{y_i y_i}(0, \left(\frac{N_2-1}{2}\right)^{\lceil}) \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ r_{y_i y_i}(\left(\frac{N_1-1}{2}\right)^{\lceil}, -\left(\frac{N_2-1}{2}\right)^{\lceil}) & \cdots & r_{y_i y_i}(\left(\frac{N_1-1}{2}\right)^{\lceil}, 0) & \cdots & r_{y_i y_i}(\left(\frac{N_1-1}{2}\right)^{\lceil}, \left(\frac{N_2-1}{2}\right)^{\lceil}) \end{bmatrix}}_{N_1 \times N_2 \text{ matrix}} \tag{3.55}$$

and

$$\mathbf{R}_{\hat{x} y_i} = \underbrace{\begin{bmatrix} r_{\hat{x} y_i}(-\left(\frac{N_1-1}{2}\right)^{\lceil}, -\left(\frac{N_2-1}{2}\right)^{\lceil}) & \cdots & r_{x y_i}(-\left(\frac{N_1-1}{2}\right)^{\lceil}, 0) & \cdots & r_{x y_i}(-\left(\frac{N_1-1}{2}\right)^{\lceil}, \left(\frac{N_2-1}{2}\right)^{\lceil}) \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ r_{\hat{x} y_i}(0, -\left(\frac{N_2-1}{2}\right)^{\lceil}) & \cdots & r_{x y_i}(0,0) & \cdots & r_{x y_i}(0, \left(\frac{N_2-1}{2}\right)^{\lceil}) \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ r_{\hat{x} y_i}(\left(\frac{N_1-1}{2}\right)^{\lceil}, -\left(\frac{N_2-1}{2}\right)^{\lceil}) & \cdots & r_{x y_i}(\left(\frac{N_1-1}{2}\right)^{\lceil}, 0) & \cdots & r_{x y_i}(\left(\frac{N_1-1}{2}\right)^{\lceil}, \left(\frac{N_2-1}{2}\right)^{\lceil}) \end{bmatrix}}_{N_1 \times N_2 \text{ matrix}}. \tag{3.56}$$

Substituting (3.50), (3.53) and (3.54) into (3.49), we have,

$$\frac{\delta J}{\delta \bar{\mathbf{f}}_i} = -2\, vec\{\mathbf{R}_{\hat{x} y_i}\} + 2\, BC\{\mathbf{R}_{y_i y_i}\}\bar{\mathbf{f}}_i + \sum_{j \neq i} BC\{\mathbf{R}_{y_i y_j}\}\bar{\mathbf{f}}_j, \tag{3.57}$$

which can be written in terms of the 2D-DFT, $\mathrm{F}\{\cdot\}$ and its inverse $\mathrm{F}^{-1}\{\cdot\}$ as shown below,

$$\frac{\delta J}{\delta \bar{\mathbf{f}}_i} = vec\left\{-2\mathbf{R}_{\hat{x}y_i} + 2\mathrm{F}^{-1}\left\{\mathrm{F}\left\{\mathbf{R}_{y_iy_i}\right\}\mathrm{F}\left\{\bar{\mathbf{F}}_i\right\}\right\} + \sum_{j\neq i}\mathrm{F}^{-1}\left\{\mathrm{F}\left\{\mathbf{R}_{y_iy_j}\right\}\mathrm{F}\left\{\bar{\mathbf{F}}_j\right\}\right\}\right\}. \tag{3.58}$$

From (3.58), the gradient needed for the steepest descent approach involves a few forward and backward $N_1 \times N_2$ 2D-DFTs of the correlation matrices. However, the gradient that we really want is the gradient of the cost, J, with respect to $\vec{f}_i$, $\frac{\delta J}{\delta \vec{f}_i}$, with no zero-padding and not $\frac{\delta J}{\delta \bar{\mathbf{f}}_i}$. We incorporate the independence between the input of the equalization filter, $y_i(n_1,n_2)$, and the error between the deconvolution output and the Bussgang nonlinearity output, $e(n_1,n_2)$ into the steepest descent algorithm, that is, $E\{y_i(n_1-k_1,n_2-k_2)e(n_1,n_2)^*\}=0$, $\forall k_1,k_2$. As a result, only the center $(2D_1-1)\times(2D_2-1)$ terms of the correlation matrices are needed. By reflecting this fact into (3.58), the expression becomes

$$\frac{\delta J}{\delta \vec{f}_i} = vec\left\{C_{D_1,D_2}\left\{-2\tilde{\mathbf{R}}_{\hat{x}y_i} + 2\mathrm{S}_{\left(\frac{D_1}{2}\right)^{\urcorner},\left(\frac{D_2}{2}\right)^{\urcorner}}\left\{\mathrm{F}^{-1}\left\{\mathrm{F}\left\{\tilde{\mathbf{R}}_{y_iy_i}\right\}\mathrm{F}\left\{\tilde{\mathbf{F}}_i\right\}\right\} + \sum_{j\neq i}\mathrm{S}_{\left(\frac{D_1}{2}\right)^{\urcorner},\left(\frac{D_2}{2}\right)^{\urcorner}}\left\{\mathrm{F}^{-1}\left\{\mathrm{F}\left\{\tilde{\mathbf{R}}_{y_iy_j}\right\}\mathrm{F}\left\{\tilde{\mathbf{F}}_j\right\}\right\}\right\}\right\}\right\} \tag{3.59}$$

where $C_{D_1,D_2}\{\cdot\}$ is an operator to extract the center $D_1 \times D_2$ submatrix, and $\mathrm{S}_{\left(\frac{D_1}{2}\right)^{\urcorner},\left(\frac{D_2}{2}\right)^{\urcorner}}\{\cdot\}$ performs a down and right circular shift of $\left(\frac{D_1}{2}\right)^{\urcorner}\times\left(\frac{D_2}{2}\right)^{\urcorner}$. The correlation matrices, $\tilde{\mathbf{R}}_{\hat{x}y_i}$, $\tilde{\mathbf{R}}_{y_iy_i}$ and $\tilde{\mathbf{R}}_{y_iy_j}$ are the center $(2D_1-1)\times(2D_2-1)$ submatrices of $\mathbf{R}_{\hat{x}y_i}$, $\mathbf{R}_{y_iy_i}$ and $\mathbf{R}_{y_iy_j}$ and finally $\tilde{\mathbf{F}}_i$ is a $(2D_1-1)\times(2D_2-1)$ defined as

$$\tilde{\mathbf{F}}_i = \underbrace{\begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & f_i(0,0) & \cdots & f_i(0,D_2-1) \\ \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & f_i(D_1-1,0) & \cdots & f_i(D_1-1,D_2-1) \end{bmatrix}}_{(2D_1-1)\times(2D_2-1)\,matrix} \tag{3.60}$$

In summary, our multi-channel equalization filter update approach as represented by equations (3.41) and (3.59) in practice requires only a few $(2D_1-1)\times(2D_2-1)$ 2D-FFTs. The order of complexity for our approach in the single channel case is

$0\{(2D_1-1)(2D_1-1)\log[(2D_1-1)(2D_1-1)]\}$ times the number of iterations, while the order of complexity in the direct method is $0\{D_1^3 D_2^3\}$. The number of iterations required in our equalization filter update approach is small (15-35) since $\hat{x}$, which is required in the gradient calculation, is itself evolving in the outer loop of the SCMB blind deconvolution algorithm. Our algorithm is also efficient in terms of memory usage: the memory storage required is $0\{(2D_1-1)(2D_2-1)\}$, while the memory storage required for the direct method is $0\{D_1^2 D_2^2\}$. Finally, the run time for a typical ISAR autofocus problem using our algorithm is about 17 times faster and the focused ISAR image is visually indistinguishable from that of the direct method.

3.8 Self-correcting multi-channel Bussgang (SCMB)

So far we have explained the key elements of the self-correcting multi-channel Bussgang (SCMB) blind deconvolution algorithm. Even though the general design philosophy should apply to all classes of signal, the actual form of the feedback and the way the input signal is modeled are application dependent. In Chapters 4 and 5, we go into more specific details of the SCMB algorithm for the binary image restoration problem and the SAR/ISAR autofocus problem respectively.

4 Binary image restoration using SCMB blind deconvolution

In this chapter, we present the details of the self-correcting multi-channel Bussgang (SCMB) blind deconvolution that were not covered in Chapter 3 and which are specific to the binary image restoration problem. They include the design of the feedback loop, modeling the probability density function (pdf), and the Bussgang nonlinearity. Simulation results applying the SCMB blind deconvolution algorithm to binary images that have been blurred by point spread functions (PSF) including out-of-focus blurs, diffusion blurs and motion blurs conclude this chapter.

4.1 Design of a feedback mechanism for SCMB blind deconvolution

The purpose of the feedback loop is to cancel the output error, $\Delta\hat{x}$, caused by the incomplete knowledge of the pdf. In the binary image case, each pixel in the image is modeled by the pdf

$$p_x(x) = p_o\delta(x) + (1-p_o)\delta(x-a), \tag{4.1}$$

where the two binary levels 0 and a correspond to the background with probability p_o and the foreground with probability $(1-p_o)$. $\delta(x)$ is the Kronecker delta function.

With the choice of a quadratic cost functional for the Bayes risk in (2.12), the binary image pdf model in (4.1) and the zero-mean white Gaussian model for the deconvolution noise, from (3.11) the Bussgang nonlinearity for binary image becomes,

$$\hat{x} = g\{\breve{x}, p_o\} = E\{x \mid \breve{x}\} = a\left[1+\left(\frac{p_o}{1-p_o}\right)e^{-\frac{a(\breve{x}-\frac{a}{2})}{\sigma_w^2}}\right]^{-1}. \tag{4.2}$$

Equation (4.2) assumes that we know p_o perfectly, which gives the correct nonlinearity output, $\hat{x}$. In reality, however, we have,

$$\hat{x} + \Delta\hat{x} = g\{\breve{x}, p_o + \Delta p_o\}, \tag{4.3}$$

where the error in the pdf parameter, Δp_o, results in the nonlinearity output error, $\Delta\hat{x}$. Therefore, we perturb the nonlinearity input $\breve{x}$ with a feedback gain, $\Delta\breve{x}$, so that the effect of the pdf parameter error will be cancelled out. Thus,

$$\hat{x} = g\{\breve{x} + \Delta\breve{x}, p_o + \Delta p_o\}. \tag{4.4}$$

The output error, $\Delta\hat{x}$, can be approximated from the partial derivatives of $g\{\cdot\}$. Setting this approximated error to zero gives

$$\Delta\hat{x} = \left[\frac{\partial g\{\breve{x}, p_o\}}{\partial\breve{x}}\right]\Delta\breve{x} + \left[\frac{\partial g\{\breve{x}, p_o\}}{\partial p_o}\right]\Delta p_o \triangleq 0. \tag{4.5}$$

The gain, $\Delta\breve{x}$, is then equal to

$$\Delta\breve{x} = -\frac{\left[\dfrac{\partial g\{\breve{x}, p_o\}}{\partial p_o}\right]}{\left[\dfrac{\partial g\{\breve{x}, p_o\}}{\partial\breve{x}}\right]}\Delta p_o. \tag{4.6}$$

Since the mean of the binary signal, $\bar{x}$, is

$$\bar{x} = a(1 - p_o), \tag{4.7}$$

the mean of the nonlinearity output, $\bar{\hat{x}}$, with the pdf parameter error becomes

$$\bar{\hat{x}} = a(1 - p_o - \Delta p_o). \tag{4.8}$$

From equations (4.7) and (4.8), we can estimate the pdf parameter error, Δp_o, from the mean of the binary image, $\bar{x}$, and the mean of the nonlinearity output, $\bar{\hat{x}}$:

$$\Delta p_o = \frac{\bar{x} - \bar{\hat{x}}}{a}. \tag{4.9}$$

By utilizing equation (4.2), the partial derivatives of $g\{\breve{x}, p_o\}$ with respect to p_o and $\breve{x}$ are

$$\frac{\partial g\{\breve{x}, p_o\}}{\partial p_o} = -a\left[1 + \left(\frac{p_o}{1 - p_o}\right)e^{-\frac{a(\breve{x} - \frac{a}{2})}{\sigma_w^2}}\right]^{-2} e^{-\frac{a(\breve{x} - \frac{a}{2})}{\sigma_w^2}}\left(\frac{1}{(1 - p_o)^2}\right) \tag{4.10}$$

and

$$\frac{\partial g\{\breve{x}, p_o\}}{\partial \breve{x}} = -a\left[1+\left(\frac{p_o}{1-p_o}\right)e^{-\frac{a(\breve{x}-\frac{a}{2})}{\sigma_w{}^2}}\right]^{-2} e^{-\frac{a(\breve{x}-\frac{a}{2})}{\sigma_w{}^2}}\left(\frac{p_o}{1-p_o}\right)\left(-\frac{a}{\sigma_w{}^2}\right). \tag{4.11}$$

Dividing (4.10) by (4.11) gives

$$\frac{\dfrac{\partial g\{\breve{x}, p_o\}}{\partial p_o}}{\dfrac{\partial g\{\breve{x}, p_o\}}{\partial \breve{x}}} = \frac{-\sigma_w{}^2}{a\,p_o(1-p_o)}. \tag{4.12}$$

It is important to realize that the ratio of the derivatives in equation (4.12) does not depend on the deconvolved ouput, $\breve{x}$, which will vary as the Bussgang algorithm iterates. This implies that the approach should be robust and independent of the initial guess.

Substituting equations (4.9) and (4.12) into (4.6) gives

$$\Delta\breve{x} = \left(\frac{\sigma_w{}^2}{a\,p_o(1-p_o)}\right)\left(\frac{\bar{x}-\bar{\hat{x}}}{a}\right). \tag{4.13}$$

By recognizing the variance of the binary signal, $\sigma_x{}^2$, as,

$$\sigma_x{}^2 = a^2 p_o(1-p_o), \tag{4.14}$$

we substitute equation (4.14) into (4.13) to get,

$$\Delta\breve{x} = \left(\frac{\sigma_w{}^2}{\sigma_x{}^2}\right)\left(\bar{x}-\bar{\hat{x}}\right). \tag{4.15}$$

The signal-to-deconvolution-noise ratio, SDNR is defined as

$$SDNR = \frac{\sigma_x{}^2}{\sigma_w{}^2}. \tag{4.16}$$

Notice that the feedback gain equation (4.15) can then be rewritten as

$$\Delta\breve{x} = \left(\frac{\bar{x}-\bar{\hat{x}}}{SDNR}\right). \tag{4.17}$$

Since the $SDNR$ is unknown and may vary from iteration to iteration, we replace $SDNR$ by $1/k$ and incorporate the iteration number, i. To improve the appearance of the result, we also switch the symbol that we used for the mean from an overbar to $E\{\cdot\}$. With these changes, the feedback gain equation becomes

$$\Delta\breve{x}^{(i)} = k(E\{x(\cdot)\} - E\{\hat{x}(\cdot)^{(i-1)}\}) . \tag{4.18}$$

Note that $\Delta\breve{x}^{(i)}$ is a constant gain for the current frame (iteration), i, which requires the expected value of the output signal, $E\{\hat{x}(\cdot)^{(i-1)}\}$ from the previous frame (iteration). Even though the form of the feedback is known at this point, the gain, k, is not yet known and the expectation of the input (original) signal $E\{x(\cdot)\}$ is not accessible.

Before we address these issues, it is helpful to look at how the feedback loop gain in (4.18) fits into the rest of the Bussgang blind deconvolution framework. To better understand the problem, we reorganize the block diagram in Figure 3-1 slightly in order to facilitate the analysis in light of control theory, as depicted in Figure 4-1. However, because the closed-loop transfer function involves mixing stream (samples within a frame) and frame (iteration) information and simultaneously updating the nonlinearity and equalization filters, the feedback loop design is a difficult and unconventional task.

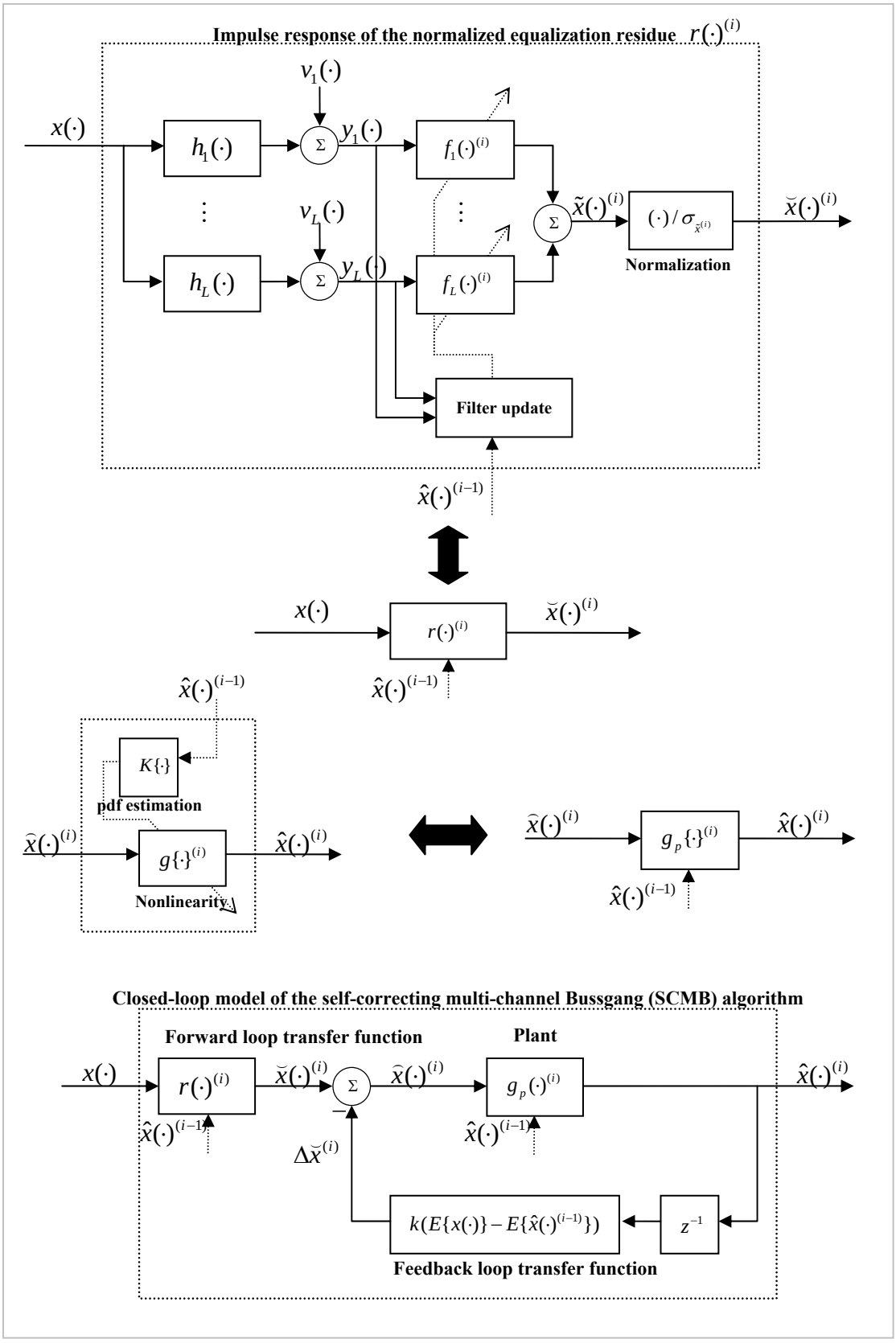


Figure 4-1 Self-correcting multi-channel Bussgang (SCMB) blind deconvolution algorithm in closed loop form

With the help of the closed-loop model shown in Figure 4-1, we continue with the design using a control theory context. The self-correcting multi-channel Bussgang (SCMB) algorithm is a discrete dynamic system in terms of the iteration number, where the plant is frame varying and nonlinear. The feedback loop gain $\Delta\breve{x}^{(i)}$ in (4.18), can be seen in the steady state to satisfy the relation

$$\lim_{i\to\infty} E\{\hat{x}(\cdot)^{(i)}\} = E\{x(\cdot)\}\,. \tag{4.19}$$

Therefore, the steady-state feedback loop gain, $\Delta\breve{x}^{(\infty)}$, is equal to zero. Thus, in the steady state the closed-loop system becomes an open-loop system and the self-correcting multi-channel Bussgang (SCMB) algorithm becomes equivalent to the original multi-channel Bussgang algorithm.

As mentioned earlier, even though the form of the feedback loop is clear, the gain k and $E\{x(\cdot)\}$ are still unknown. By exploiting the fact that the deconvolution noise, $w(\cdot)$, is zero mean, and assuming that the nonlinearity $g\{\cdot\}$ has little affect on the mean at convergence, that is $E\{\tilde{x}(\cdot)\} \approx E\{\hat{x}(\cdot)\}$ at convergence, we conclude that

$$k \approx 1\,. \tag{4.20}$$

The derivation of this result is based on Figure 4-1. The deconvolution noise, $w(\cdot)$ is known to be a zero-mean white Gaussian sequence, which by definition, is given by

$$w(\cdot) = \tilde{x}(\cdot) - x(\cdot)\,. \tag{4.21}$$

Since the output signal of the forward loop $\breve{x}(\cdot) = \tilde{x}(\cdot)/\sigma_{\tilde{x}}$ is, in fact, the normalized deconvolution output, the normalized deconvolution noise

$$v(\cdot) \triangleq \breve{x}(\cdot) - x(\cdot)\,, \tag{4.22}$$

is also zero-mean white and Gaussian. From Figure 4-1, the input to the plant (Bussgang nonlinearity), $\hat{x}(\cdot)$, can be expressed as

$$\hat{x}(\cdot)^{(i)} = \breve{x}(\cdot)^{(i)} - k\left(E\{x(\cdot)\} - E\{\hat{x}(\cdot)^{(i-1)}\}\right). \tag{4.23}$$

By substituting (4.23) into (4.22) and adding the frame (iteration) index i to (4.22), we have

$$v(\cdot)^{(i)} = \hat{x}(\cdot)^{(i)} + k\left(E\{x(\cdot)\} - E\{\hat{x}(\cdot)^{(i-1)}\}\right) - x(\cdot)\,. \tag{4.24}$$

By taking the expected values of both sides of this equation at convergence, we have

$$E\{v(\cdot)^{(\infty)}\} = E\{\hat{x}(\cdot)^{(\infty)}\} + k\left(E\{x(\cdot)\} - E\{\hat{x}(\cdot)^{(\infty)}\}\right) - E\{x(\cdot)\}\,. \tag{4.25}$$

By exploiting the fact that $v(\cdot)$ has a zero mean and rearranging equation (4.25), we get

$$0 = E\{\hat{x}(\cdot)^{(\infty)}\} - k\,E\{\hat{x}(\cdot)^{(\infty)}\} + (k-1)\,E\{x(\cdot)\}\,. \tag{4.26}$$

Therefore, with the assumption that the nonlinearity $g\{\cdot\}$ does not change the mean value significiantly at convergence, $E\{\hat{x}(\cdot)^{(\infty)}\} \approx E\{\hat{x}(\cdot)^{(\infty)}\}$, from (4.26) we conclude that $k \approx 1$ as stated in (4.20).

Note that the impulse response of the normalized equalization residue $r(\cdot)^{(i)}$ in the system is unknown and frame varying. There is no known control solution for this system. However, with the aid of methods like root locus and using the concept of envelope design, we can control the dynamic performance of the closed-loop system in Figure 4-1 so that the system is stable and converges quickly to the solution. The resulting design gives the optimal value of k to be $k = 1.2$.

In Figure 4-2 below, we give an example of the convergence of the algorithm for different choices of the feedback gain, k, ranging from 0.7 to 1.8. For feedback gains between 0.8 and 1.5, the system is stable and converges to the highest signal-to-estimated-deconvolution-noise ratio (SEDNR). Also, in Figure 4-3, we show the convergence of the background probability, p_o, for different choices of the feedback gain. Again for feedback gains between 0.8 and 1.5, the background probability, p_o, converges to the correct value of 0.913. This confirms that our choice of the feedback gain of $k = 1.2$ is reasonable.

Now the unknown gain k has been resolved, in order to finish the feedback loop design, we need to solve for the unknown expectation of the input (original) signal, $E\{x(\cdot)\}$. When the expectation of the input signal is expected to be small, we can just set it to zero and let the feedback loop account for it. Otherwise, we can get an estimate of $E\{x(\cdot)\}$ using the Gaussian mixture model and expectation maximization (EM) algorithm to estimate the parameters of the pdf for updating the nonlinearity $g\{\cdot\}^{(i)}$.

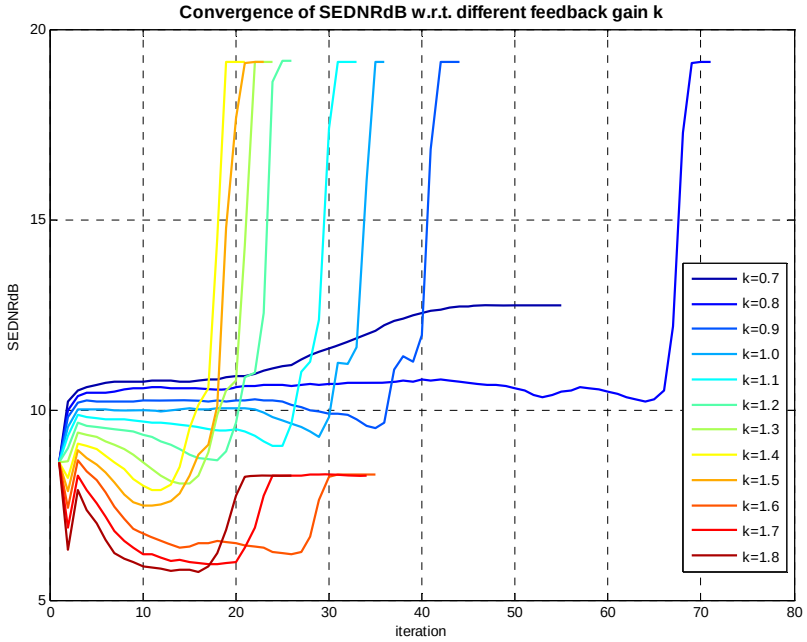


Figure 4-2 Convergence of the signal-to-estimated-deonvolution-noise ratio (SEDNR) for different choices of the feedback gain.

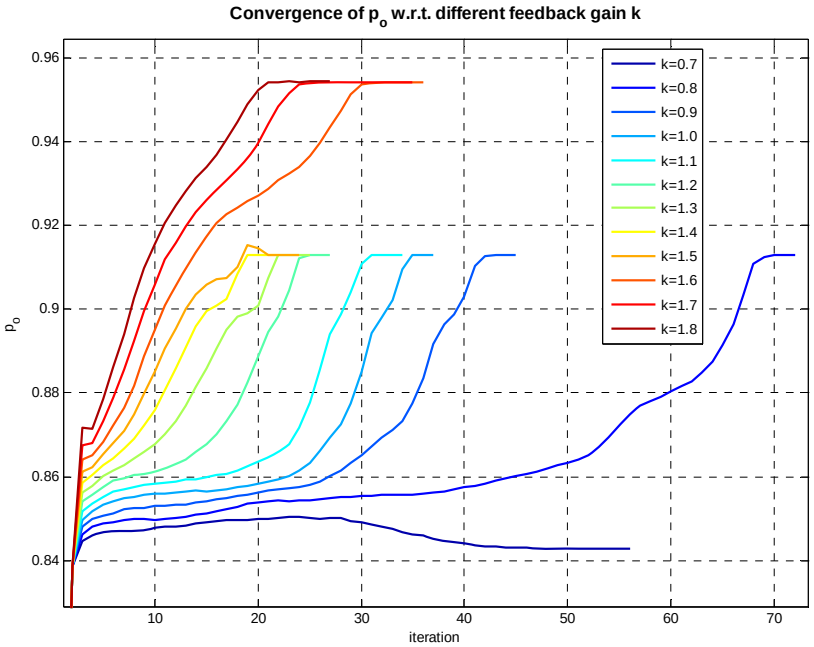


Figure 4-3 Convergence of the background probability for different choices of the feedback gain.

4.2 Gaussian mixture pdf modeling and estimation using the EM algorithm

In the binary image case, the binary pixel value, $x(\cdot)$, is real and can be modeled by a general two-component Gaussian mixture,

$$p_x(x) = \sum_{c=1}^{2} w_c \left[\left(\frac{1}{\sqrt{2\pi}\,\sigma_c} e^{-\frac{(x-\mu_c)^2}{2\sigma_c^2}} \right) \right], \tag{4.27}$$

instead of the more obvious binary model used in (4.1). This is because the two-component Gaussian mixture pdf model can fit both the perfect and distorted binary images while the other cannot. As described in Section 2.4, the expectation maximization algorithm is used to estimate the parameters of the Gaussian mixture model so that the Bussgang nonlinearity, $g\{\cdot\}$, can be updated.

4.3 Nonlinearity of SCMB for binary images

The general form of the nonlinearity of the SCMB algorithm is given in (3.11) for an input signal with an arbitrary distribution. For a binary signal the pdf is given by

$$p_x(x) = p_o\,\delta(x) + (1-p_o)\,\delta(x-a), \tag{4.28}$$

where p_o is the probability of $x=0$ (the background) and $(1-p_o)$ is the probability of $x=a$ (the foreground).

By using the zero-mean white Gaussian assumption for the deconvolution noise $w(\cdot)$,

$$p_w(w) = \frac{1}{\sqrt{2\pi}\,\sigma_w} e^{\frac{1}{2}\left(\frac{w^2}{\sigma_w^2}\right)}, \tag{4.29}$$

the nonlinearity for the binary signal becomes

$$g\{\breve{x}\} = \left(\frac{\sigma_x}{\sqrt{p_o(1-p_o)}} \right) \left[1 + \left(\frac{p_o}{1-p_o} \right) e^{-\left(\frac{\frac{\breve{x}}{\sigma_x} - \frac{1}{2\sqrt{p_o(1-p_o)}}}{\frac{\sqrt{p_o(1-p_o)}}{SDNR}} \right)} \right]^{-1}, \tag{4.30}$$

where σ_x is the standard deviation of the input signal. Because of the normalization in the functional block diagram of the SCMB algorithm shown in Figure 3-1, we can set $\sigma_x = 1$. The signal to deconvolution noise ratio is defined as $SDNR = \frac{\sigma_x^2}{\sigma_w^2} = \frac{1}{\sigma_w^2}$ which reflects how good the solution will be and how well the Bussgang process assumption holds. We typically set $SDNR = 100000 = 50\,\text{dB}$.

4.4 Simulation results

In this section, we experiment with the design of the self-correcting multi-channel Bussgang (SCMB) blind deconvolution algorithm for binary images. The original undistorted binary image, the "Text image", used in the experiment is shown in Figure 4-4 and the sensitivity of the SCMB algorithm with respect to the blurred signal to noise ratio (BSNR) for binary image restoration is shown in Figure 4-5 and Figure 4-6 below.

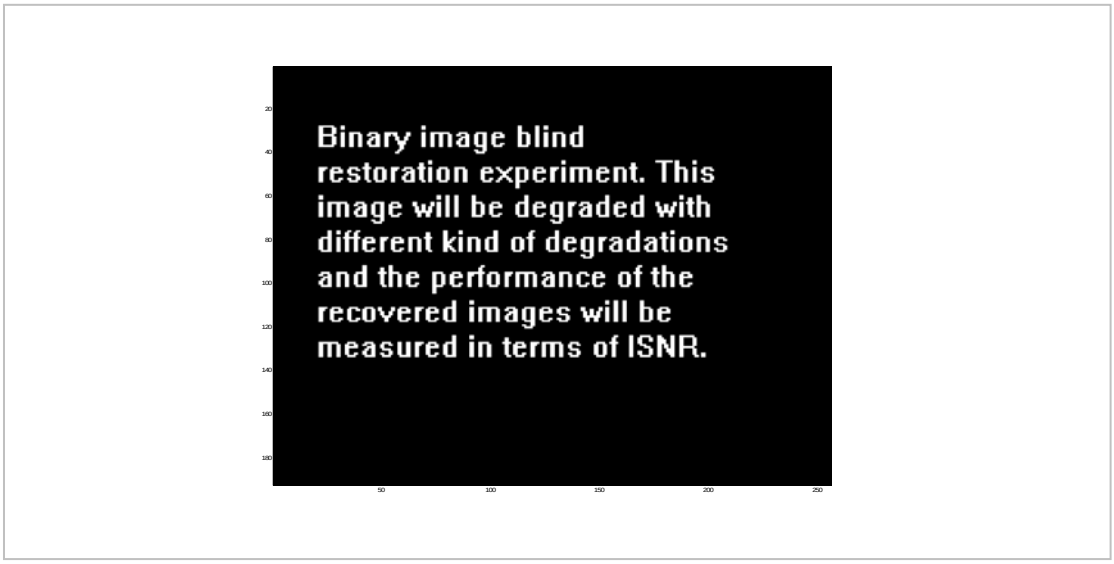
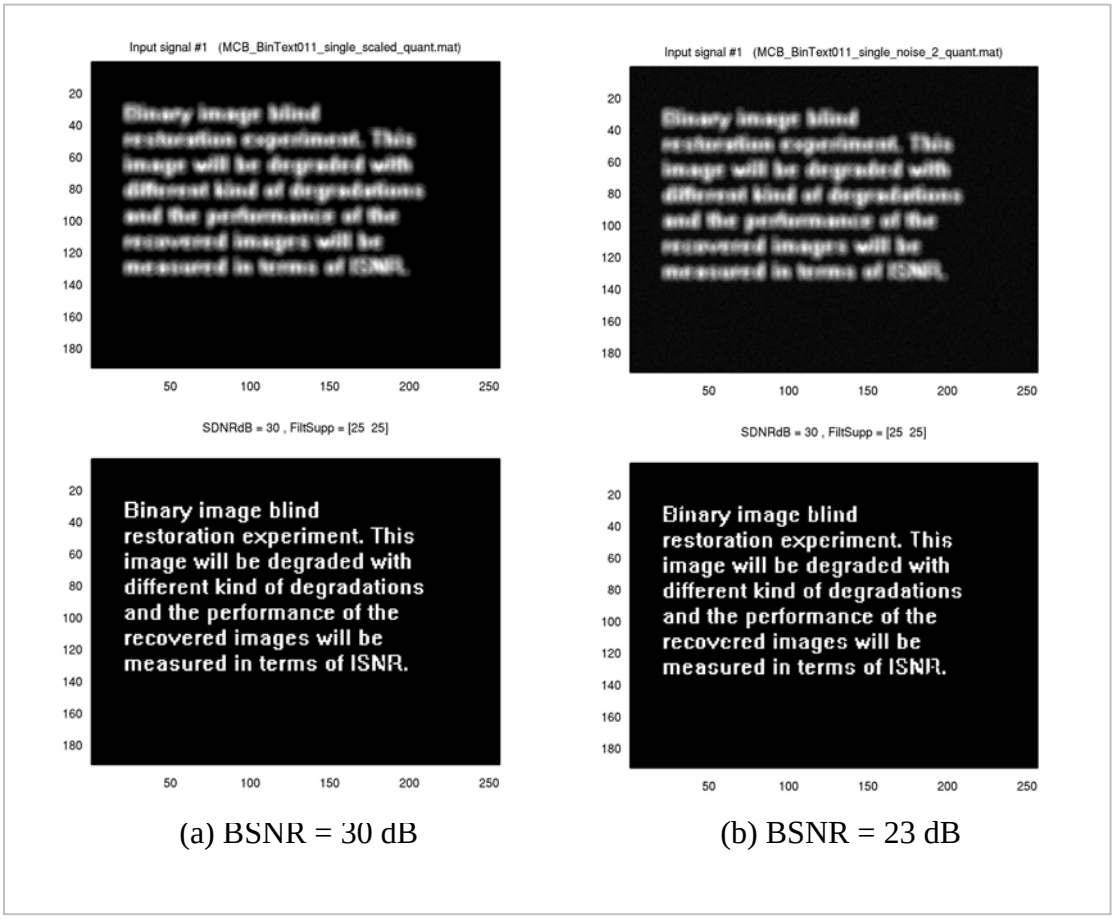
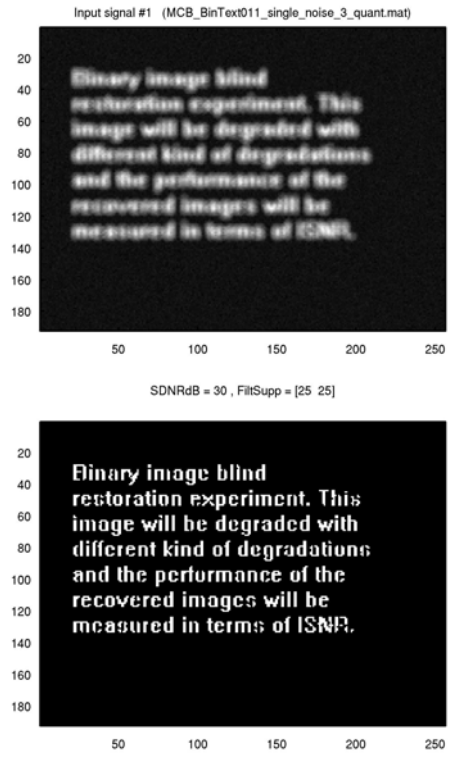


Figure 4-4 The original binary image used in the experiment, the Text image.

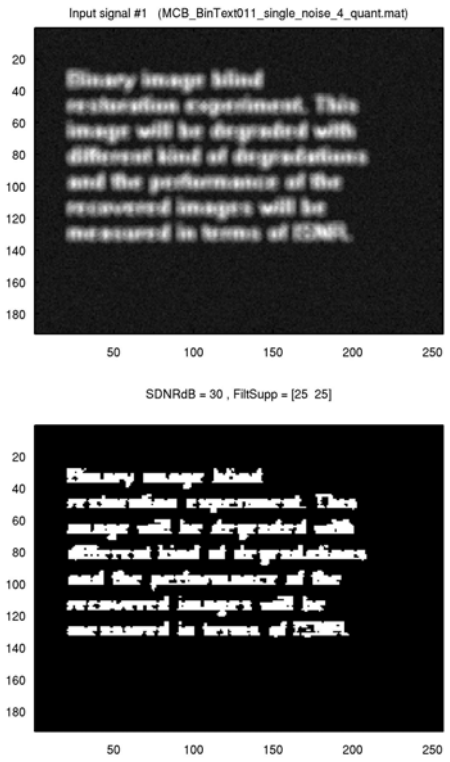


(a) BSNR = 30 dB (b) BSNR = 23 dB

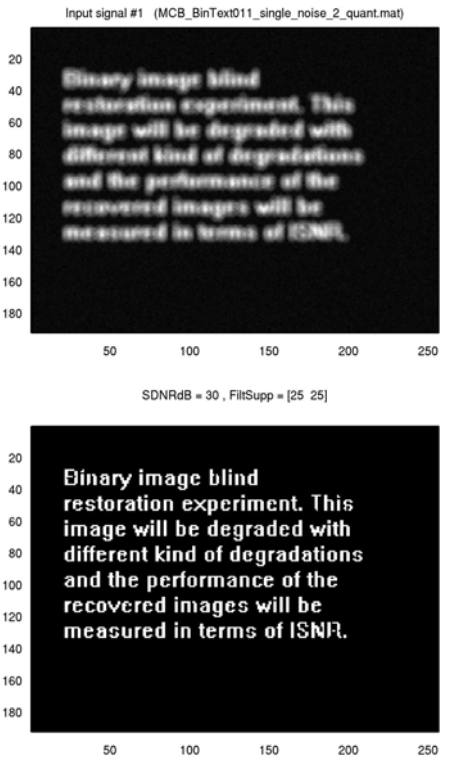
Figure 4-5 Sensitivity of SCMB to BSNR for binary images



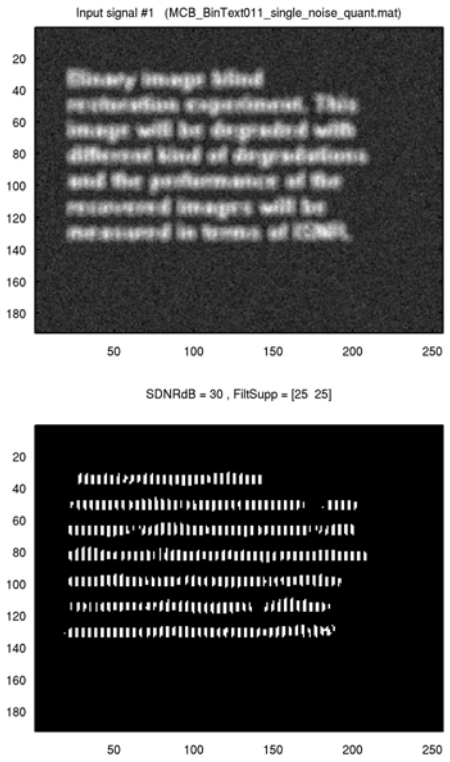
(c) BSNR = 16 dB



(d) BSNR = 14dB



(e) BSNR = 12 dB



(f) BSNR = 10 dB

Figure 4-6 Sensitivity of SCMB to BSNR for binary images (continue)

We conclude that, when the blurred binary images have a BSNR of 16 dB or greater, the SCMB algorithm is extraordinarily robust and capable of dealing with exceptionally severe blurs. The SCMB algorithm can recover information from seriously blurred images that are visually unrecognizable. In the interest of brevity, we show experiments with only two representative binary images here, namely the "Text image" (which has already been used in the example in Section 3.2) and the printed circuit broad image, "PCB image" shown in Figure 4-7 below.

In the experiments that we report here, we always process the whole image as shown in Figure 4-7, but when showing the results, we often prefer to display multiple zoomed portions of the recovered image to clearly demonstrate the performance of the algorithm. Furthermore, we use a metric that we call the signal-to-estimated-deconvolution-noise ratio (SEDNR) to measure the convergence of the SCMB algorithm. That ratio is

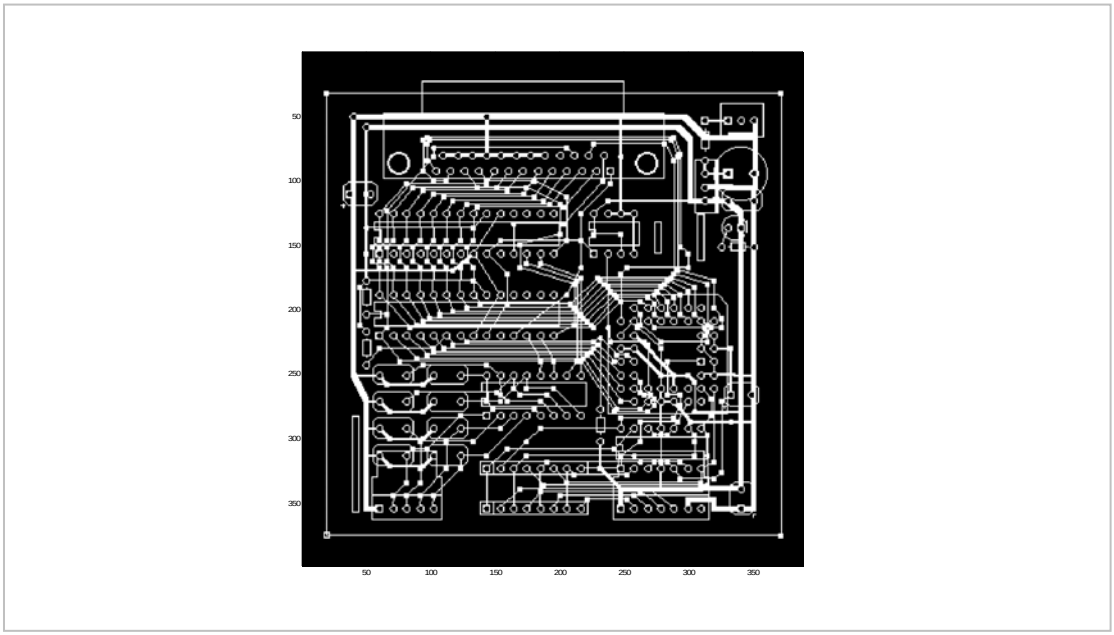


Figure 4-7 PCB image

$$SEDNR \triangleq \frac{\sigma_{\hat{x}}^2}{\sigma_{\varepsilon}^2}, \tag{4.31}$$

where $\sigma_{\hat{x}}$ is the standard deviation of the nonlinearity output and σ_{ε} is the standard deviation of the estimated deconvolution noise $\varepsilon(\cdot)$ as defined in (3.12). This metric can

also reflect the performance of the SCMB algorithm. Typically with $SEDNR \geq 8$ the recovered image is visually almost indistinguishable from the original image.

The simulation results on the "Text image" and "PCB image" are shown respectively in Table 4-1 and Table 4-2. The "Text image" is chosen to represent a type of binary image that has a high percentage of background pixels or a small mean value $E\{x(\cdot)\}$. Therefore, as described in Section 3.3, we do not estimate $E\{x(\cdot)\}$ and instead set $E\{x(\cdot)\} = 0$ in the feedback structure. The "PCB image", however, represents a binary image that does not fit that small mean value approximation and requires the explicit estimation of the mean $E\{x(\cdot)\}$ at each iteration. In our experiments, we use the common optical point spread functions defined in Section 2.6.2 for the testing. They are the out-of-focus (pillbox) blur, the diffusion (Gaussian) blur and the motion (directional) blur with support sizes as large as 17x17. The support size of the equalization filter is typically 25x25 but a maximum support size of 31x31 is used in one case. In any case, the SCMB algorithm should work with even higher support sizes for both the optical point spread function and the equalization filter; what interests us particularly is the readability of the image before and after the restoration under the severely blurred condition.

While it is not possible to show all the restored images used to generate Table 4-1 andTable 4-2, the blurred and restored images in Experiment #4 and Experiment #11 are shown in Figure 4-9 and Figure 4-10, respectively. In both cases, the recovered images are visually flawless. In addition, the estimated probability of background p_o and the mean $E\{x(\cdot)\}$ are correct despite the severity of the blur.

By comparing the result of Experiment #5 with #6, it is evident that the multi-channel case outperforms the single channel case. Experiment #6 is identical to Experiment #5 except that channel #2 does not exist. The multi-channel case in Experiment #5 converges faster, has a significantly better estimate of the background probability p_o and a much better estimate of the deconvolution noise ratio (SEDNR) than its single channel counterpart.

Another important observation can be made by comparing Experiment #11 with Experiment #12. Experiment #11 is identical to Experiment #12 except that the support size of the equalization filter is significantly larger. For experiment #11, the support size of the equalization filter is 31x31 while for Experiment #12 it is 25x25. Although in Experiment #11, convergence requires more iterations to converge because of the 672 extra filter coefficients that need to be estimated, the algorithm nonetheless manages to converge to the globally optimum solution. The equalization filters for both Experiment #11 and #12 are shown in Figure 4-8.

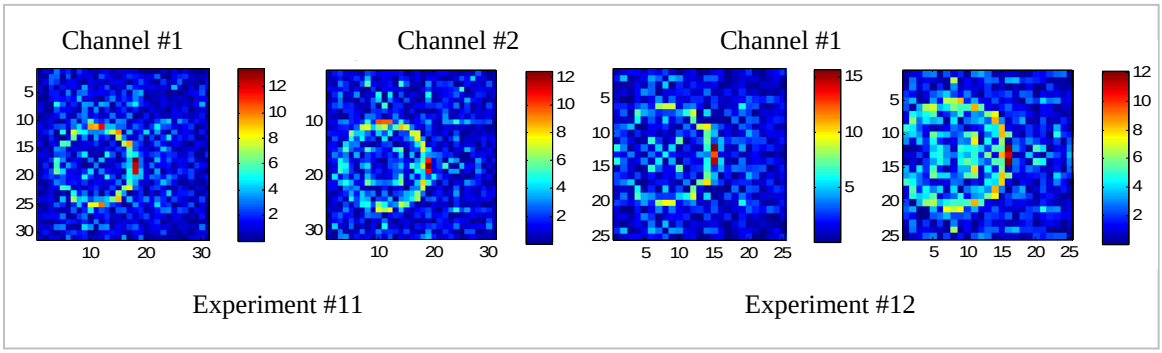


Figure 4-8 Equalization filters of Experiments #11 and #12

Experiment number		Initial guess of the background probability p_o	Initial guess of the mean of the input data $E\{x(\cdot)\}$	Channel #1 Type of impulse response	Channel #1 Support of impulse response	Channel #2 Type of impulse response	Channel #2 Support of impulse response	Support of equalization filter for all channel	Number of iterations needed for convergence	Estimated background probability p_o	Estimated mean of the input data $E\{x(\cdot)\}$	Signal to estimated deconvolution noise ratio (SEDNR) in dB
Text image	1	0.8	0	Motion blur @ 134^o	11x11	Motion blur @ 135^o	13x13	25x25	28	0.913	NA	20.54
	2	0.8	0	Diffusion blur	15x7	Diffusion blur	7x15	25x25	28	0.914	NA	12.03
	3	0.8	0	Out of focus blur	11x11	Out of focus blur	9x9	25x25	22	0.913	NA	18.51
	4	0.8	0	Out of focus blur	15x15	Out of focus blur	11x11	25x25	28	0.913	NA	17.15
	5	0.8	0	Out of focus blur	13x13	Motion blur @ 6^o	15x15	25x25	41	0.913	NA	14.22
	6	0.8	0	Out of focus blur	13x13	NA	NA	25x25	63	0.936	NA	5.88

Table 4-1 Simulation results of SCMB algorithm for Text image (k=1.2)

Experiment number		Initial guess of the background probability p_o	Initial guess of the mean of the input data $E\{x(\cdot)\}$	Channel #1 Type of impulse response	Channel #1 Support of impulse response	Channel #2 Type of impulse response	Channel #2 Support of impulse response	Support of equalization filter for all channel	Number of iterations needed for convergence	Estimated background probability p_o	Estimated mean of the input data $E\{x(\cdot)\}$	Signal to estimated deconvolution noise ratio (SEDNR) in dB
PCB image	7	0.97	0	Out of focus blur	19x11	Motion blur @ 22^o	17x17	25x25	42	0.8	0.5	11.93
	8	0.97	0	Out of focus blur	15x15	Diffusion blur	13x13	25x25	18	0.797	0.5042	8.26
	9	0.97	0	Diffusion blur	9x9	Diffusion blur	13x13	25x25	18	0.8	0.4999	14.60
	10	0.97	0	Motion blur @ 125^o	15x15	Motion blur @ 120^o	17x17	25x25	49	0.8	0.4999	18.72
	11	0.97	0	Out of focus blur	15x15	Out of focus blur	17x17	31x31	51	0.8	0.4999	15.24
	12	0.97	0	Out of focus blur	15x15	Out of focus blur	17x17	25x25	31	0.8	0.4999	14.15

Table 4-2 Simulation results of SCMB algorithm on PCB image (k=1.2)

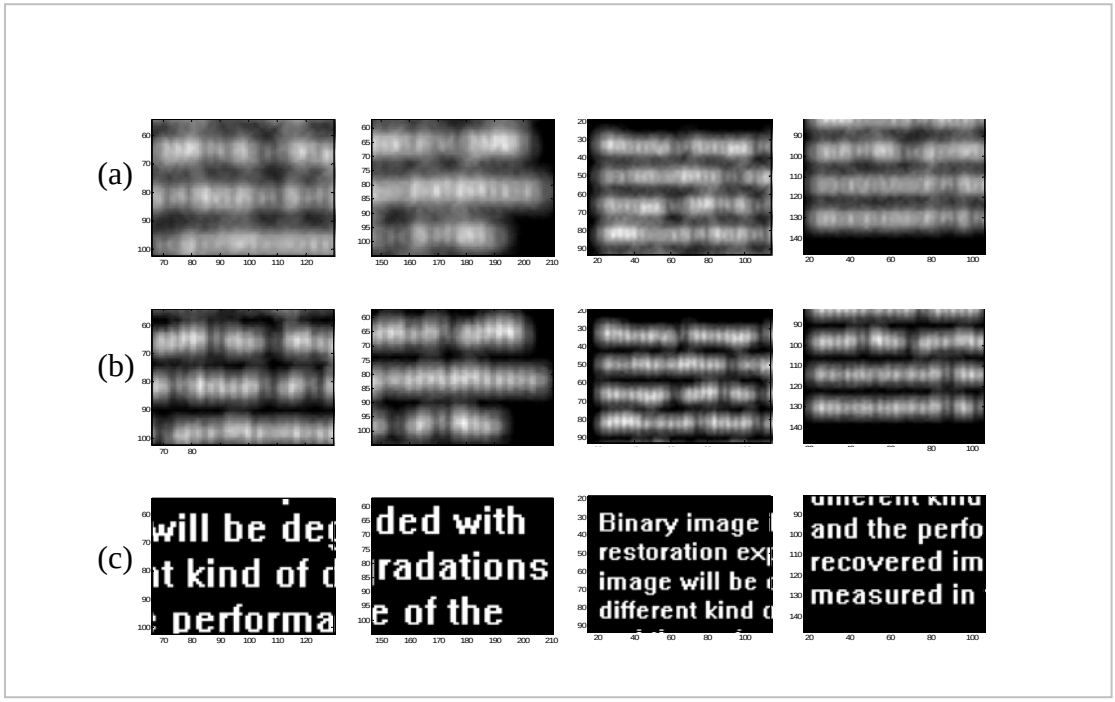


Figure 4-9 Blurred text image and SCMB results at different zoom levels. (a) [15x15] Out-of-focus blur in channel #1, (b) [11x11] Out-of-focus blur in channel #2, (c) SCMB blind deconvolution output

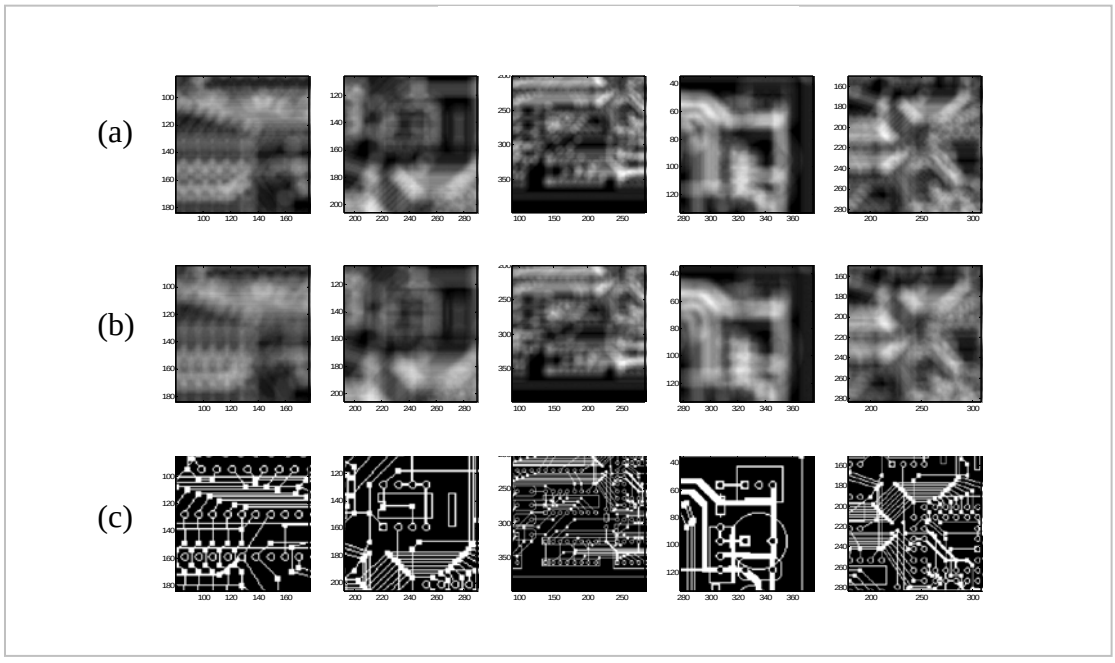


Figure 4-10 Blurred PCB image and SCMB results at different zoom levels. (a) [15x15] Out-of-focus blur in channel #1, (b) [17x17] Out-of-focus blur in channel #2, (c) SCMB blind deconvolution output

5 SAR/ISAR Autofocus using SCMB Blind Deconvolution

In this chapter, we present the details of the self-correcting multi-channel Bussgang (SCMB) blind deconvolution algorithm that were not covered in chapter 3 and which are specific to the SAR/ISAR autofocus problem. They are the feedback loop design, the modeling of the probability density function (pdf), and the selection of the Bussgang nonlinearity. We would like to point out that even though there are few differences in the design of the SCMB blind deconvolution algorithm for the SAR and ISAR autofocus problems, our primary interest is in the ISAR autofocus problem simply because ISAR autofocus techniques that are applicable to scenarios with significant non-uniform rotational motion estimation error are few and ineffective. Furthermore, the PSF model for the ISAR autofocus problem as described in (2.37) fits quite well with the two-dimensional non-separable complex equalization filters in the SCMB blind deconvolution algorithm. Since the PSF model for the SAR autofocus problem in (2.36) is a special case of the PSF model for the ISAR autofocus problem, the methodology should work for the SAR autofocus problem as well.

5.1 Gaussian mixture pdf modeling and estimation using the EM algorithm

To implement the Bussgang nonlinearity, the complex gray-level SAR signal is broken down into two parts: the natural logarithm of the amplitude, $\hat{z}(\cdot)$, and the phase, $\hat{\theta}(\cdot)$, as shown in Figure 5-1,

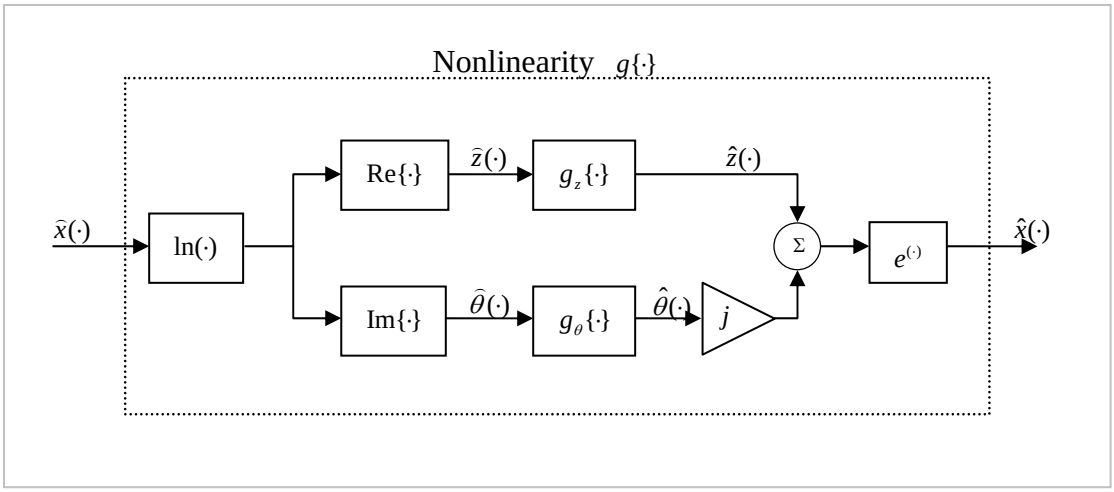


Figure 5-1 Non-linearity for SAR image

where $\ln(\cdot)$, $\mathrm{Re}\{\cdot\}$, $\mathrm{Im}\{\cdot\}$, j and $e^{(\cdot)}$ are, respectively, the natural logarithm function, real part operator, imaginary part operator, $\sqrt{-1}$ and the exponential function.

In the SAR imaging case, the input signal $x(\cdot) = a(\cdot)e^{j\theta(\cdot)}$ is complex. The pdf of the phase is modeled by a uniform distribution between 0 and 2π, without any unknown modeling parameters as shown below

$$p_\theta(\theta) = \frac{1}{2\pi}, \qquad 0 \le \theta \le 2\pi \,. \tag{5.1}$$

The pdf of the natural logarithm of the amplitude, $z(\cdot) \triangleq \ln(a(\cdot))$, is modeled using a three-component Gaussian mixture model,

$$p_z(z) = \sum_{c=1}^{3} w_c \left[\left(\frac{1}{\sqrt{2\pi}\,\sigma_c} e^{-\frac{(z-\mu_c)^2}{2\sigma_c^2}} \right) \right]. \tag{5.2}$$

It is the error in the estimation of the Gaussain mixture parameters (the weights, w_i, the means, μ_i and the standard deviations, σ_i, where i is the Gaussian component number) that the feedback loop is designed to remove.

Because of the way the input signals are modeled, the nonlinearity $g\{\cdot\}$ decomposes into the nonlinearities $g_z\{\cdot\}$ and $g_\theta\{\cdot\}$, respectively, for the natural logarithm of the amplitude and the phase of the input signal. Their relationship is as shown in Figure 5-1.

5.2 Nonlinearity of SCMB for SAR/ISAR images

The general form of the nonlinearity of the SCMB algorithm was given in (3.11), where the nonlinearity output signal $g(\breve{x})$ is the conditional expectation of the unknown input signal x given the deconvolved output signal $\breve{x}$. In the SAR/ISAR imaging case, the signal is complex and can be modeled by a complex Gaussian distribution. This is equivalent to modeling the magnitude by a Rayleigh distribution and the phase by a uniform distribution. However, neither a Gaussian nor a complex Gaussian input signal is allowed in the Bussgang technique. Thus, instead of using the complex Gaussian model for the input signal x, we break down the nonlinearity into the two components as shown in Figure 5-1. We use a Gaussian mixture distribution to model the magnitude of the input signal instead of a Rayleigh distribution while retaining the uniform distribution model for the phase of the input signal. This avoids the widely used complex Gaussian model for the SAR/ISAR signal, which would result in a linear relationship between the deconvolved output signal $\breve{x}$ and the nonlinearity output signal $g(\breve{x})$, and violate the nonlinearity assumption in the Bussgang framework. The performance of the resulting system justifies our model.

With the magnitude of the input signal modeled by a three-component Gaussian mixture as shown in (5.2) and the phase modeled by a uniform distribution as shown in (5.1), and by defining the deconvolution noises for both magnitude and phase components respectively as

$$w_z = \hat{z} - z\,, \tag{5.3}$$

and

$$w_\theta = \hat{\theta} - \theta\,, \tag{5.4}$$

where both deconvolution noises are assumed to be white and Gaussian distributed, we use the Bussgang nonlinearity expressed in terms of the pdf as in (3.11) and apply it to both the magnitude and phase signals. The derivation of the nonlinearity of the magnitude signal begins below.

Applying the nonlinearity in (3.11) to the natural logarithm of the amplitude of x, z, as shown in Figure 5-1, we have

$$g_z\{\hat{z}\} \triangleq E\{z \mid \hat{z}\} = \frac{\int z\, p_z(z)\, p_{w_z}(\hat{z}-z)\, dz}{\int p_z(z)\, p_{w_z}(\hat{z}-z)\, dz}\,. \tag{5.5}$$

The pdf of the three-component Gaussian mixtures is

$$p_z(z) = \sum_{i=1}^{3} \left(\frac{w_i}{\sqrt{2\pi}\,\sigma_i} \right) e^{-\frac{(z-\mu_i)^2}{2\sigma_i^2}} \tag{5.6}$$

and the pdf for the zero-mean, white Gaussian deconvolution noise for the magnitude is

$$p_{w_z}(w_z) = \left(\frac{1}{\sqrt{2\pi}\,\sigma_{w_z}} \right) e^{-\frac{w_z^2}{2\sigma_{w_z}^2}}\,. \tag{5.7}$$

Substituting the definition of the deconvolution noise for the magnitude from (5.3) into (5.7), we have

$$p_{w_z}(\hat{z}-z) = \left(\frac{1}{\sqrt{2\pi}\,\sigma_{w_z}} \right) e^{-\frac{(\hat{z}-z)^2}{2\sigma_{w_z}^2}}\,. \tag{5.8}$$

Combining (5.6) and (5.8), the product of probabilities in the integrands of (5.5) becomes

$$p_z(z)\, p_{w_z}(\hat{z}-z) = \sum_{i=1}^{3}\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right) e^{-\frac{(z-\mu_i)^2}{2\,\sigma_i^{\,2}}}\, e^{-\frac{(\hat{z}-z)^2}{2\sigma_{w_z}^{\,2}}}. \tag{5.9}$$

Now, substituting (5.9) into the amplitude nonlinearity $g_z(\hat{z})$ equation in (5.5), we have

$$\hat{z} = g_z\{\hat{z}\} = \frac{\int z \sum_{i=1}^{3}\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right) e^{-\frac{(z-\mu_i)^2}{2\,\sigma_i^{\,2}}}\, e^{-\frac{(\hat{z}-z)^2}{2\sigma_{w_z}^{\,2}}}\, dz}{\int \sum_{i=1}^{3}\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right) e^{-\frac{(z-\mu_i)^2}{2\,\sigma_i^{\,2}}}\, e^{-\frac{(\hat{z}-z)^2}{2\sigma_{w_z}^{\,2}}}\, dz}. \tag{5.10}$$

We reorganize the expression in (5.10) by pulling the summation outside the integrals and combining the exponential terms. Then,

$$\hat{z} = \frac{\sum_{i=1}^{3}\left(\int z\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right) e^{-\frac{1}{2}\left[\frac{(z-\mu_i)^2}{\sigma_i^{\,2}}+\frac{(\hat{z}-z)^2}{\sigma_{w_z}^{\,2}}\right]} dz\right)}{\sum_{i=1}^{3}\left(\int\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right) e^{-\frac{1}{2}\left[\frac{(z-\mu_i)^2}{\sigma_i^{\,2}}+\frac{(\hat{z}-z)^2}{\sigma_{w_z}^{\,2}}\right]} dz\right)}. \tag{5.11}$$

To simplify the expression in (5.11) (specifically to remove the integration in both the numerator and the denominator), we reorganize the powers of the common exponential term inside the square brackets above by completing the square as

$$\left[\frac{(z-\mu_i)^2}{\sigma_i^{\,2}}+\frac{(\hat{z}-z)^2}{\sigma_{w_z}^{\,2}}\right] = \frac{(z-a_i)^2+c_i}{b_i^{\,2}}, \tag{5.12}$$

where $a_i = \dfrac{\sigma_{w_z}^{\,2}\mu_i+\sigma_i^{\,2}\hat{z}}{\sigma_{w_z}^{\,2}+\sigma_i^{\,2}}$, $b_i^{\,2} = \dfrac{\sigma_i^{\,2}\sigma_{w_z}^{\,2}}{\sigma_{w_z}^{\,2}+\sigma_i^{\,2}}$, and $c_i = \dfrac{\sigma_{w_z}^{\,2}\mu_i^{\,2}+\sigma_i^{\,2}\hat{z}^2}{\sigma_{w_z}^{\,2}+\sigma_i^{\,2}} - \left(\dfrac{\sigma_{w_z}^{\,2}\mu_i+\sigma_i^{\,2}\hat{z}}{\sigma_{w_z}^{\,2}+\sigma_i^{\,2}}\right)^2$.

Using (5.12), we can evaluate the integral in the denominator of (5.11). We begin with

$$\int\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right) e^{-\frac{1}{2}\left[\frac{(z-\mu_i)^2}{\sigma_i^{\,2}}+\frac{(\hat{z}-z)^2}{\sigma_{w_z}^{\,2}}\right]} dz = \int\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right) e^{-\frac{1}{2}\left[\frac{(z-a_i)^2+c_i}{b_i^{\,2}}\right]} dz. \tag{5.13}$$

We now try to write the integral as an integral of the pdf of a Gaussian distribution. This gives

$$\int\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right)e^{-\frac{1}{2}\left[\frac{(z-\mu_i)^2}{\sigma_i^2}+\frac{(\tilde{z}-z)^2}{\sigma_{w_z}^2}\right]}dz=\left(\frac{w_i\,b_i\,e^{-\frac{1}{2}\left(\frac{c_i}{b_i^2}\right)}}{\sqrt{2\pi}\,\sigma_{w_z}\,\sigma_i}\right)\int\frac{1}{\sqrt{2\pi}\,b_i}e^{-\frac{1}{2}\frac{(z-a_i)^2}{b_i^2}}dz\,. \tag{5.14}$$

Since the remaining integral is equal to 1, we have

$$\int\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right)e^{-\frac{1}{2}\left[\frac{(z-\mu_i)^2}{\sigma_i^2}+\frac{(\tilde{z}-z)^2}{\sigma_{w_z}^2}\right]}dz=\left(\frac{w_i\,b_i\,e^{-\frac{1}{2}\left(\frac{c_i}{b_i^2}\right)}}{\sqrt{2\pi}\,\sigma_{w_z}\,\sigma_i}\right)\,. \tag{5.15}$$

Similarly we can write the integral in the numerator of (5.11) as

$$\int z\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right)e^{-\frac{1}{2}\left[\frac{(z-\mu_i)^2}{\sigma_i^2}+\frac{(\tilde{z}-z)^2}{\sigma_{w_z}^2}\right]}dz=\left(\frac{w_i\,b_i\,e^{-\frac{1}{2}\left(\frac{c_i}{b_i^2}\right)}}{\sqrt{2\pi}\,\sigma_{w_z}\,\sigma_i}\right)\int z\frac{1}{\sqrt{2\pi}\,b_i}e^{-\frac{1}{2}\frac{(z-a_i)^2}{b_i^2}}dz\,. \tag{5.16}$$

Notice that the integral on the right-hand side of (5.16) is the first moment or the mean of z with $p_z(z)=\frac{1}{\sqrt{2\pi}\,b_i}e^{-\frac{1}{2}\frac{(z-a_i)^2}{b_i^2}}$. Therefore,

$$\int z\frac{1}{\sqrt{2\pi}\,b_i}e^{-\frac{1}{2}\frac{(z-a_i)^2}{b_i^2}}dz=a_i\,. \tag{5.17}$$

Substituting (5.17) into (5.16), we have

$$\int z\left(\frac{w_i}{2\pi\,\sigma_{w_z}\,\sigma_i}\right)e^{-\frac{1}{2}\left[\frac{(z-\mu_i)^2}{\sigma_i^2}+\frac{(\tilde{z}-z)^2}{\sigma_{w_z}^2}\right]}dz=\left(\frac{w_i\,b_i\,e^{-\frac{1}{2}\left(\frac{c_i}{b_i^2}\right)}a_i}{\sqrt{2\pi}\,\sigma_{w_z}\,\sigma_i}\right)\,. \tag{5.18}$$

Now, substituting (5.18) and (5.15) into (5.11), we get

$$\hat{z} = \frac{\sum_{i=1}^{3}\left(a_i \frac{w_i b_i e^{-\frac{1}{2}\left(\frac{c_i}{b_i^2}\right)}}{\sqrt{2\pi}\,\sigma_{w_z}\sigma_i} \right)}{\sum_{i=1}^{3}\left(\frac{w_i b_i e^{-\frac{1}{2}\left(\frac{c_i}{b_i^2}\right)}}{\sqrt{2\pi}\,\sigma_{w_z}\sigma_i} \right)}. \tag{5.19}$$

Finally, the expression for the nonlinearity of the amplitude is obtained by further simplifying (5.19) by writing out b_i explicitly and by defining $d_i = \frac{c_i}{b_i^2}$, such that

$$\hat{z} = \frac{\sum_{i=1}^{3}\left(a_i \frac{w_i e^{-\left(\frac{1}{2}\right)d_i}}{\sqrt{2\pi\left(\sigma_{w_z}{}^2 + \sigma_i^2\right)}} \right)}{\sum_{i=1}^{3}\left(\frac{w_i e^{-\left(\frac{1}{2}\right)d_i}}{\sqrt{2\pi\left(\sigma_{w_z}{}^2 + \sigma_i^2\right)}} \right)}, \tag{5.20}$$

where $a_i = \frac{\sigma_{w_z}{}^2 \mu_i + \sigma_i^2 \hat{z}}{\sigma_{w_z}{}^2 + \sigma_i^2}$, $d_i = \frac{\left(\hat{z} - \mu_i\right)^2}{\sigma_{w_z}{}^2 + \sigma_i^2}$, and σ_{w_z} is the standard deviation of the deconvolution noise for the magnitude of the input signal. w_i, σ_i and μ_i are, respectively, the weight, standard deviation and mean of the ith Gaussian component.

For the nonlinearity of the phase, θ, shown in Figure 5-1, we have

$$g_\theta\{\hat{\theta}\} \triangleq E\{\theta \mid \hat{\theta}\} = \frac{\int \theta\, p_\theta(\theta)\, p_{w_\theta}(\hat{\theta} - \theta)\, d\theta}{\int p_\theta(\theta)\, p_{w_\theta}(\hat{\theta} - \theta)\, d\theta}. \tag{5.21}$$

As already mentioned, the pdf of the phase is modeled as a uniform distribution between $-\pi$ and π, i.e.,

$$p_\theta(\theta) = \frac{1}{2\pi}, \qquad -\pi \le \theta \le \pi. \tag{5.22}$$

The pdf for the assumed zero-mean white Gaussian deconvolution noise for the phase is

$$p_{w_\theta}(w_\theta) = \left(\frac{1}{\sqrt{2\pi}\,\sigma_{w_\theta}} \right) e^{-\frac{w_\theta^2}{2\sigma_{w_\theta}{}^2}}. \tag{5.23}$$

Substituting the definition the deconvolution noise for the phase from (5.4) into (5.23), we have

$$p_{w_\theta}(\hat{\theta}-\theta)=\left(\frac{1}{\sqrt{2\pi}\,\sigma_{w_\theta}}\right)e^{-\frac{(\hat{\theta}-\theta)^2}{2\sigma_{w_\theta}{}^2}}. \tag{5.24}$$

Now, substituting (5.22) and (5.24) into (5.21) gives

$$\hat{\theta}=g_\theta\{\hat{\theta}\}=\frac{\int_{-\pi}^{\pi}\theta\, e^{-\frac{(\hat{\theta}-\theta)^2}{2\sigma_{w_\theta}{}^2}}d\theta}{\int_{-\pi}^{\pi}e^{-\frac{(\hat{\theta}-\theta)^2}{2\sigma_{w_\theta}{}^2}}d\theta}. \tag{5.25}$$

We can simplify this expression by applying the identities below, respectively, to the numerator and the denominator with $a=\hat{\theta}$ and $b=\sqrt{2}\,\sigma_{w_\theta}$:

$$\int_{-\pi}^{\pi}u\,e^{-\frac{(a-u)^2}{b^2}}du=\left(\frac{\sqrt{\pi}}{2}\right)b\,a\left[erf\left(\frac{a+\pi}{b}\right)-erf\left(\frac{a-\pi}{b}\right)\right]+\frac{b^2}{2}\left[e^{-\left(\frac{a+\pi}{b}\right)^2}-e^{-\left(\frac{a-\pi}{b}\right)^2}\right], \tag{5.26}$$

$$\int_{-\pi}^{\pi}e^{-\frac{(a-u)^2}{b^2}}du=\left(\frac{\sqrt{\pi}}{2}\right)b\left[erf\left(\frac{a+\pi}{b}\right)-erf\left(\frac{a-\pi}{b}\right)\right], \tag{5.27}$$

Here $erf(u)=\frac{2}{\sqrt{\pi}}\int_0^u e^{-t^2}dt$ is the odd error function, such that, $erf(-u)=-erf(u)$. The final form for the phase nonlinearity becomes

$$\hat{\theta}=\hat{\theta}+\frac{2\sigma_{w_\theta}\left(e^{\frac{-(\pi+\hat{\theta})^2}{2\sigma_{w_\theta}{}^2}}-e^{\frac{-(\pi-\hat{\theta})^2}{2\sigma_{w_\theta}{}^2}}\right)}{\sqrt{2\pi}\left(erf\left(\frac{\pi+\hat{\theta}}{\sqrt{2}\,\sigma_{w_\theta}}\right)+erf\left(\frac{\pi-\hat{\theta}}{\sqrt{2}\,\sigma_{w_\theta}}\right)\right)} \tag{5.28}$$

The derivations of the identities (5.26) and (5.27) are given in Appendix **Error! Reference source not found.**.

5.3 Design of a feedback mechanism for SCMB blind deconvolution

The feedback design for the SAR imaging case follows the same philosophy as in the binary imaging case. It tries to cancel the nonlinearity output error caused by the inaccurate estimation of the pdf of the input data or errors in the estimated parameters of the pdf model by altering the input to the nonlinearity with feedback. Although the philosophy behind the feedback loop design is very similar for both the binary and the SAR imaging case, the feedback design for the SAR imaging case is significantly more complicated than its binary imaging counterpart.

We begin by looking at the relationship between the error of the nonlinearity output, $\Delta\hat{z}$, with respect to errors of the Gaussian mixture parameters, Δw_i, $\Delta\mu_i$, $\Delta\sigma_i$ and the change of the nonlinearity input, $\Delta\hat{z}$. To a first order approximation, this relationship is given by the partial differential equation,

$$\Delta\hat{z} = \sum_{i=1}^{3}\left\{\left[\frac{\partial\hat{z}}{\partial w_i}\right]\Delta w_i + \left[\frac{\partial\hat{z}}{\partial\mu_i}\right]\Delta\mu_i + \left[\frac{\partial\hat{z}}{\partial\sigma_i}\right]\Delta\sigma_i\right\} + \left[\frac{\partial\hat{z}}{\partial\hat{z}}\right]\Delta\hat{z}\,. \tag{5.29}$$

We then force this error, $\Delta\hat{z}$, to zero by perturbing the nonlinearity input $\Delta\hat{z}$ (the feedback signal) as follows:

$$\Delta\hat{z} = -\sum_{i=1}^{3}\left\{\frac{\left[\frac{\partial\hat{z}}{\partial w_i}\right]}{\left[\frac{\partial\hat{z}}{\partial\hat{z}}\right]}\Delta w_i + \frac{\left[\frac{\partial\hat{z}}{\partial\mu_i}\right]}{\left[\frac{\partial\hat{z}}{\partial\hat{z}}\right]}\Delta\mu_i + \frac{\left[\frac{\partial\hat{z}}{\partial\sigma_i}\right]}{\left[\frac{\partial\hat{z}}{\partial\hat{z}}\right]}\Delta\sigma_i\right\}. \tag{5.30}$$

The partial derivatives $\frac{\partial\hat{z}}{\partial w_i}$, $\frac{\partial\hat{z}}{\partial\mu_i}$, $\frac{\partial\hat{z}}{\partial\sigma_i}$ and $\frac{\partial\hat{z}}{\partial\hat{z}}$ can be obtained by taking the derivatives of (5.20) with respect to w_i, μ_i, σ_i and $\hat{z}$ respectively. These partial derivatives will depend on the nonlinearity input, $\hat{z}$. Therefore, the feedback gain will be a function of the nonlinearity input, $\hat{z}$, and will not be a constant feedback gain as it was on the binary imaging case. Although these partial derivatives are realizable, the expressions and their derivations are quite involved. Therefore, Mathematica was used for their derivation and the results are presented separately in Appendix **Error! Reference source not found.**.

Even with the partial derivatives in (5.30) available, we still need to estimate the Gaussian mixture parameterization errors, Δw_i, $\Delta\mu_i$ and $\Delta\sigma_i$, in order to calculate the feedback gain function, $\Delta\hat{z}(\hat{z})$. We proceed with the definition of the variance of the natural logarithm of the amplitude of the SAR signal, σ_z^2, given by,

$$\sigma_z^{\,2} = \int (z - \bar{z})^2 p_z(z) dz \,. \tag{5.31}$$

By substituting (5.2) into (5.31) and using the identity

$$\int_{-\infty}^{\infty} z^2 e^{-\frac{1}{2}\left(\frac{z-\mu_i}{\sigma_i}\right)^2} dz = \sqrt{2\pi}\,\sigma_i \left(\mu_i^{\,2} + \sigma_i^{\,2}\right), \tag{5.32}$$

we get

$$\sigma_z^{\,2} + \bar{z}^2 = \sum_{i=1}^{3} w_i (\mu_i^{\,2} + \sigma_i^{\,2}) \,. \tag{5.33}$$

This relates the second-order statistics of the natural logarithm of the amplitude of the SAR signal to all of the parameters of the Gaussian mixture model with no parameterization errors.

For the non-ideal case where there are parameterization errors, the second-order statistics of the natural logarithm of the amplitude of the SAR signal are replaced by the corresponding statistics of the nonlinearity output and all of the parameters of the Gaussian mixture model are perturbed by the parameterization error. That is,

$$\sigma_{\hat{z}}^{\,2} + \bar{\hat{z}}^2 = \sum_{i=1}^{3} \left(w_i + \Delta w_i\right)\left[(\mu_i + \Delta\mu_i)^2 + (\sigma_i + \Delta\sigma_i)^2\right]. \tag{5.34}$$

By expanding the right-hand side of (5.34) and making the assumption that higher-order terms formed by the products of Δw_i, $\Delta\mu_i$ and $\Delta\sigma_i$ are insignificant and incorporating the fact that the sum of the changes in the weights is constrained to be zero

$$\sum_{i=1}^{3} \Delta w_i = 0 \,, \tag{5.35}$$

we conclude that

$$q = \mathbf{v}^T \mathbf{\Delta p} \,, \tag{5.36}$$

where

$$q \triangleq \sigma_{\hat{z}}^{\,2} + \bar{\hat{z}}^2 - \sum_{i=1}^{3} w_i (\mu_i^{\,2} + \sigma_i^{\,2}) - (\mu_3^{\,2} + \sigma_3^{\,2}) \tag{5.37}$$

$$\begin{aligned} \mathbf{v}^T \triangleq \big[& 2w_1\mu_1 \quad 2w_1\sigma_1 \quad 2w_2\mu_2 \quad 2w_2\sigma_2 \quad 2w_3\mu_3 \quad 2w_3\sigma_3 \quad \cdots \\ & \mu_1^{\,2} + \sigma_1^{\,2} - \mu_3^{\,2} - \sigma_3^{\,2} \quad \mu_2^{\,2} + \sigma_2^{\,2} - \mu_3^{\,2} - \sigma_3^{\,2} \big] \end{aligned} \tag{5.38}$$

and

$$\Delta\mathbf{p}^T \triangleq \left[\Delta\mu_1 \quad \Delta\sigma_1 \quad \Delta\mu_2 \quad \Delta\sigma_2 \quad \Delta\mu_3 \quad \Delta\sigma_3 \quad \Delta w_1 \quad \Delta w_2\right] \tag{5.39}$$

Therefore, the minimum mean-squared estimates of the Gaussian mixture parameterization errors can be found by solving

$$\Delta\mathbf{p} = \left(\mathbf{v}\,\mathbf{v}^T\right)^{-1}\mathbf{v}\,q \tag{5.40}$$

$$\Delta w_3 = -\Delta w_1 - \Delta w_2 \tag{5.41}$$

Equation (5.40) is the solution to (5.36) and (5.41) follows from (5.35). Notice that since $\left(\mathbf{v}\,\mathbf{v}^T\right)$ is of rank one, the matrix inverse in (5.40) must actually be replaced by the pseudo-inverse. Thus, the system of equations for solving for the Gaussian mixture parameterization errors is underdetermined and a unique solution does not exist. Unlike its binary imaging counterpart, this is not surprising, since the same second-order statistics for the Gaussian mixture can be realized by more than one combination of Gaussian signals when there is more than one Gaussian in the mixture. In any case, the feedback gain, $\Delta\bar{z}(\bar{z})$, for the SAR imaging case is found by using equations (5.30), (5.40) and (5.41).

5.4 Simulation results

5.4.1 SAR autofocus results

The following simulation results use the pdf model proposed for SAR as shown in Section 5.1, where the natural logarithm of the amplitude of the complex SAR signal is modeled by a three-component Gaussian mixture model and the phase is modeled by a uniform distribution model bounded between zero and 2π. The unknown parameters of the Gaussian mixture model used in the simulation are static and are extracted by employing the expectation maximization (EM)-based Gaussian mixture estimation technique as described in Section 2.4. The prototype SAR image on which the EM-based Gaussian mixture estimation technique is applied, is a synthetic SAR image generated from the back scatter of a three-dimensional CAD model by the polar format algorithm (PFA) SAR image formation technique. This prototype SAR image of a tank used in this simulation is shown in Figure 5-2. In the simulation, the proposed self-correcting multi-channel Busssgang (SCMB) blind deconvolution algorithm with the Bussgang nonlinearities as derived in Section 5.2 is applied simultaneously on two Ku-band, real SAR images with slightly different aspects. To the knowledge of the author, this is the first SAR autofocus technique that can take advantage of multiple defocused SAR images.

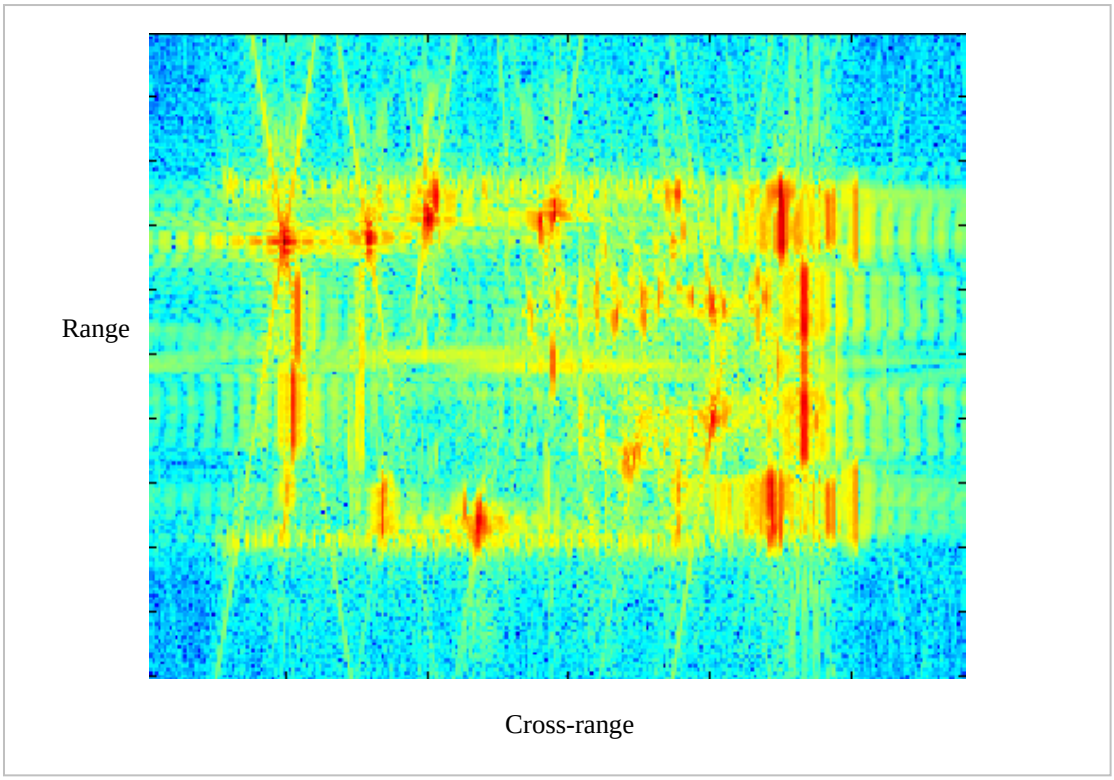


Figure 5-2 The synthetic SAR image of a tank for pdf extraction.

The defocusing of the SAR image occurs naturally in the cross-range dimension (x-axis) because of the range error from one radar pulse to another caused by the inaccuracies of the IMU and GPS units onboard. This effect is especially important in high frequency bands such as Ku-band. The two defocused SAR images are as shown below in Figure 5-3 and Figure 5-4.

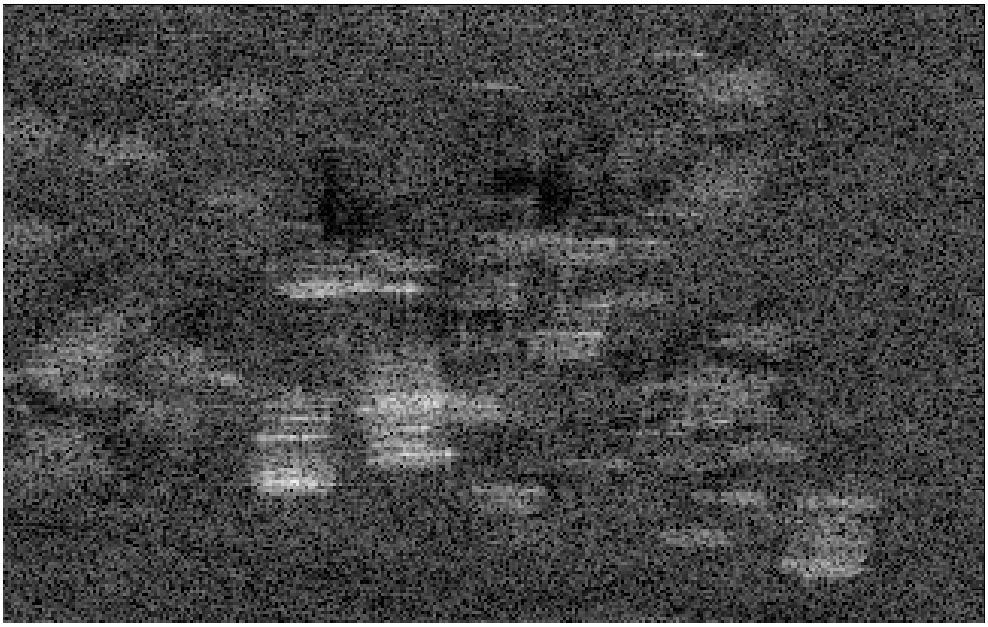


Figure 5-3 Defocused Ku-band SAR image #1.

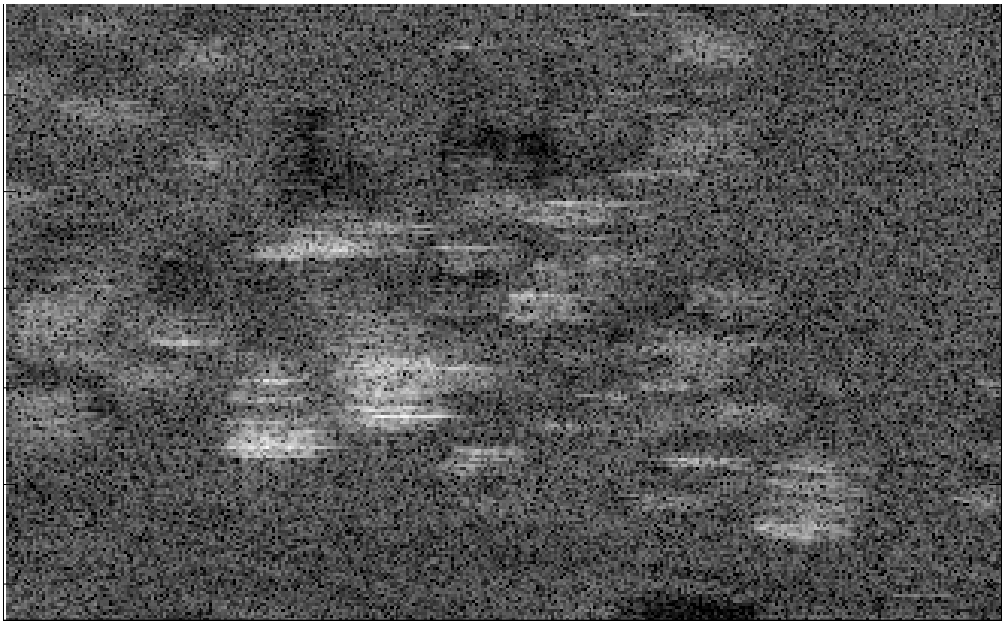


Figure 5-4 Defocused Ku-band SAR image #2.

The resulting SAR image focused by our SCMB algorithm that takes advantage of both of the defocused SAR images from Figure 5-3 and Figure 5-4 is shown in Figure 5-5. Significant improvements in terms of the focusing, the contrast, and the signal-to-noise ratio (SNR) of the resulted SAR image are observed.

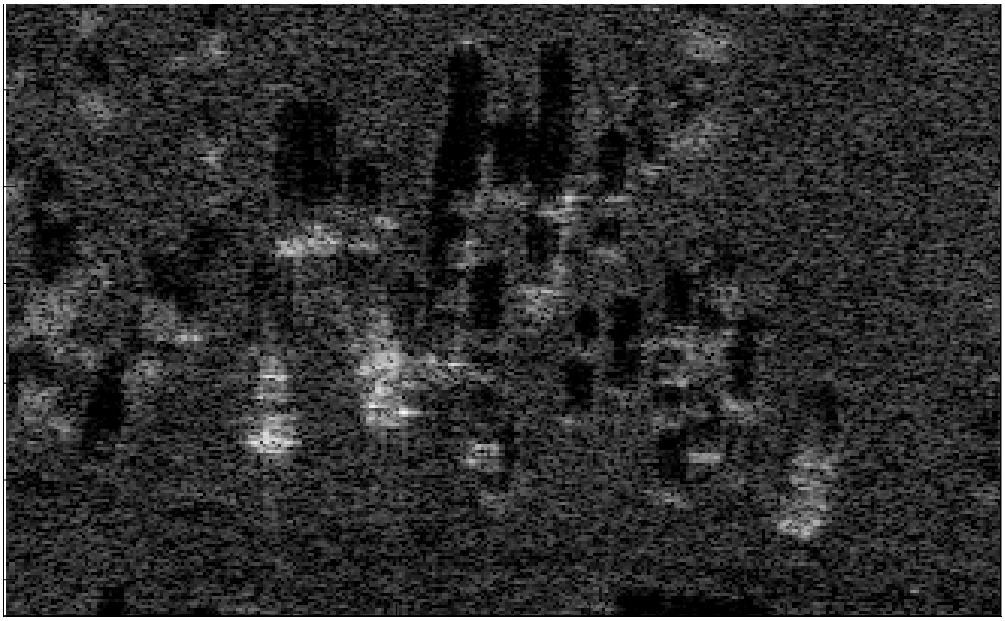


Figure 5-5 Focused Ku-band SAR image by SCMB with static prototype PDF.

It is logical to compare this result with that of the phase gradient algorithm (PGA), a well-established SAR autofocus technique often considered the gold standard for SAR autofocus. This method was described in Section 2.5.2. Since PGA can only be applied to one defocused SAR image at a time, we applied PGA to the two defocused Ku-band SAR images separately. The two focused images are shown in Figure 5-6 and Figure 5-7.

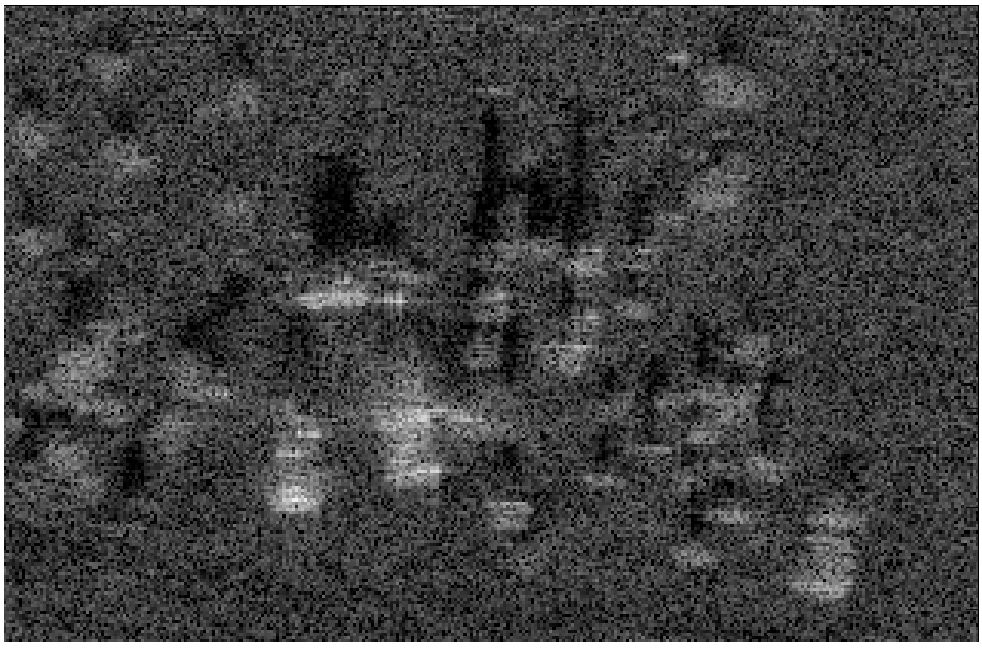


Figure 5-6 Focused image of defocused Ku-band SAR image #1 by PGA.

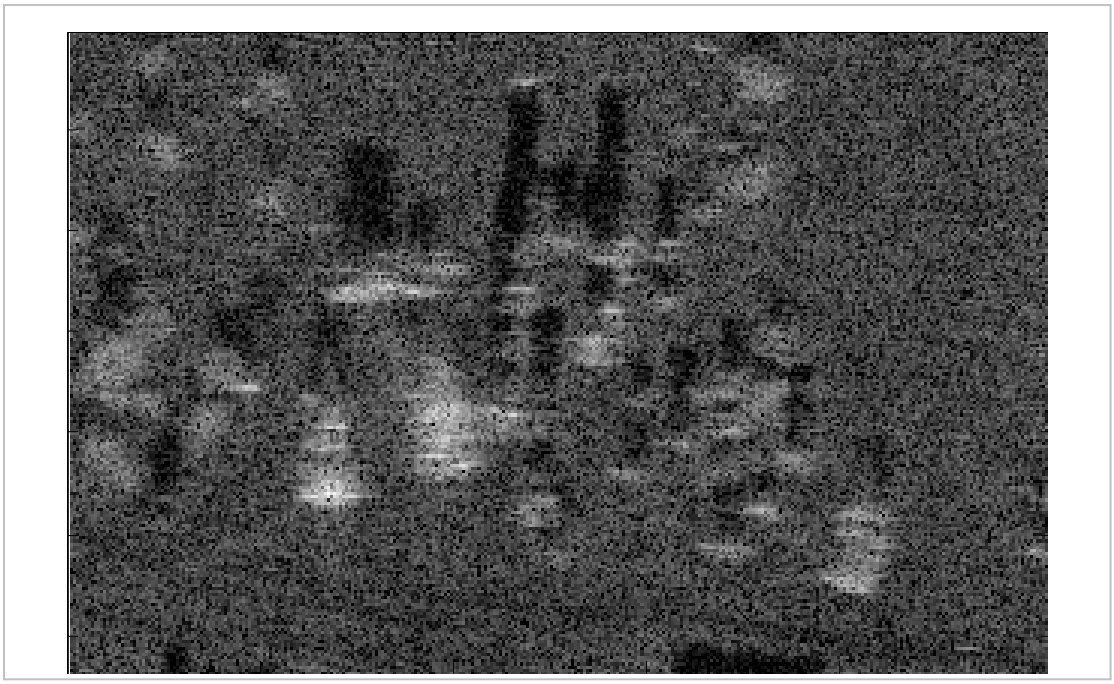


Figure 5-7 Focused image of defocused Ku-band SAR image #2 by PGA.

In comparing the PGA autofocus results in Figure 5-6 and Figure 5-7 with that of SCMB in Figure 5-5, it is evident that the SCMB blind deconvolution algorithm surpasses the performance of PGA. In the SAR image focused by SCMB, the vehicles in the scene are more pronounced with more solid edges and the prominent point scatterers on the vehicles are more noticeable. The shadows cast by the vehicles, the building materials, and the trees in the scene are also darker than on the PGA focused images.

In previous experiment we compared the two-channel SCMB blind deconvolution algorithm with the PGA autofocus algorithm. Now we want to compare the performance of SCMB blind deconvolution algorithm against itself. We zoom in on one of the vehicles from the scene in the previous simulation and compare the autofocus performance of the single-channel, two-channel and three-channel versions of the SCMB. The results are shown in Figure 5-8, where (a) is the defocused image used for the single-channel case, (a) and (b) were used for the two-channel case, (a), (b) and (c) we used in the three-channel case. The autofocus results of the single-channel, two-channel and three-channel SCMB autofocusing are shown respectively, in (d), (e) and (f). We conclude from this experiment that the two-channel case focused better than the single-channel case, while the three-channel case is visually indistinguishable from the two-channel case.

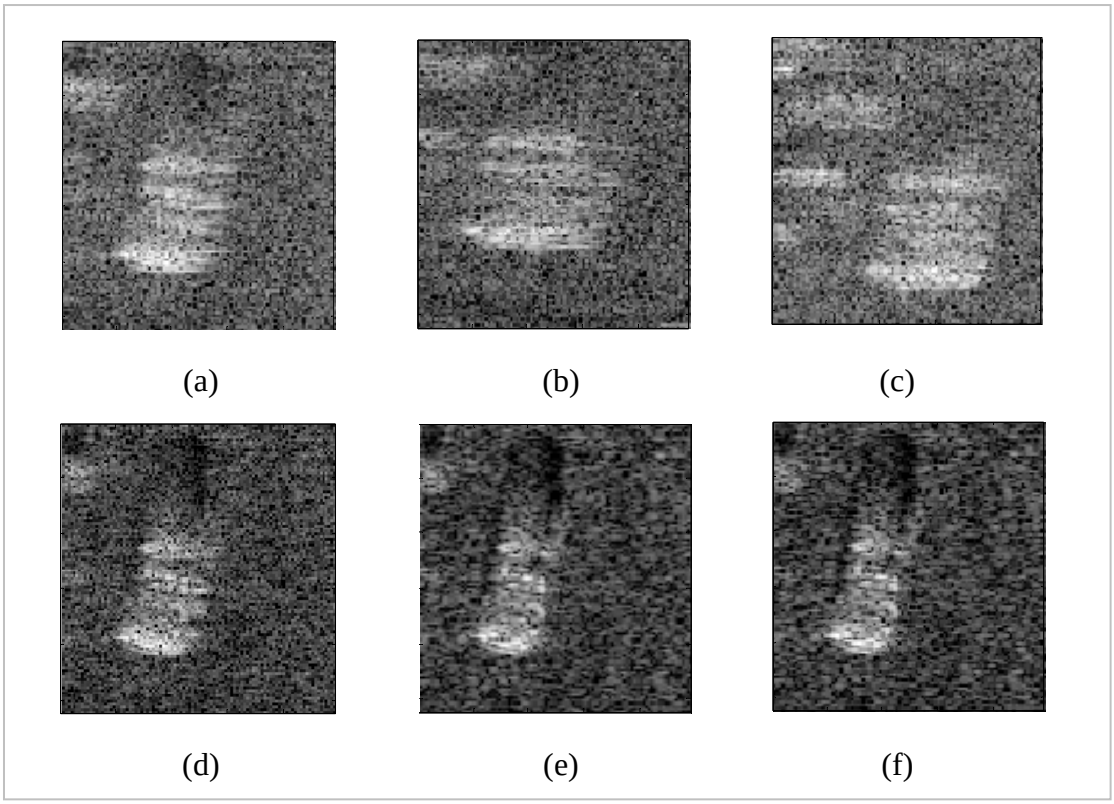


Figure 5-8 Focused Ku-band SAR images by SCMB with static prototype PDF, (a), (b), (c) are the defocused images (d) Single-channel result, (e) Two-channel result and (f) Three-channel result.

5.4.2 ISAR autofocus results

In this experiment, the SCMB algorithm was used to tackle the ISAR autofocus problem in a single channel setup. Data collection was carried out using an X-band radar mounted on a 300 feet high tower and pointing towards a moving vehicle at the scene center (the origin). The depression angle formed by the line of sight (LOS) between the radar and the scene center, at the center of the radar footprint, was about 17 degrees. The trajectory formed by the moving vehicle with the scene center at the origin is as shown in Figure 5-9.

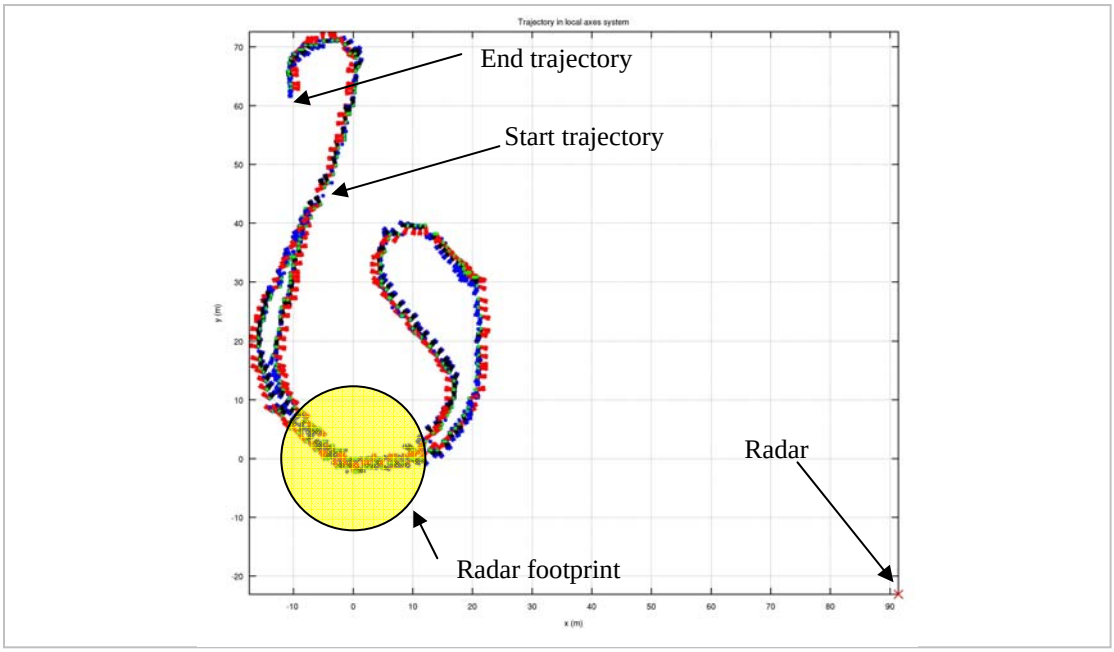


Figure 5-9 Trajectory of moving vehicle (A or B) in the ISAR scenario.

The trajectory was obtained by the GPS installed in the vehicle which had a precision of 5 meters to approximate the position of the vehicle resolved by multiple radars. Two different vehicles (A and B) were used in the experiment. The heading of the vehicle is crudely approximated by the direction of the velocity vector which is calculated from the first derivative of the position vector (the trajectory). The moving vehicle evidently entered the radar footprint, made a turn, and left the radar footprint as observed from both the trajectory and range compressed pulse diagrams respectively in Figure 5-9 and Figure 5-10. In the range compressed pulse diagram, we select the pulses bounded by the two vertical red lines which correspond to a center azimuth angle of roughly 25 degrees with respect to the vehicle's body axis.

To form an ISAR image, we first motion compensate the IQ signal to account for the difference in range between the radar and the vehicle from pulse to pulse. Since phase is a much more significant contributor than amplitude, we motion compensate by phase only for simplicity. In order to show the importance of the motion compensation step, we show the ISAR image before and after motion compensation for vehicle A, respectively in Figure 5-11 and Figure 5-12.

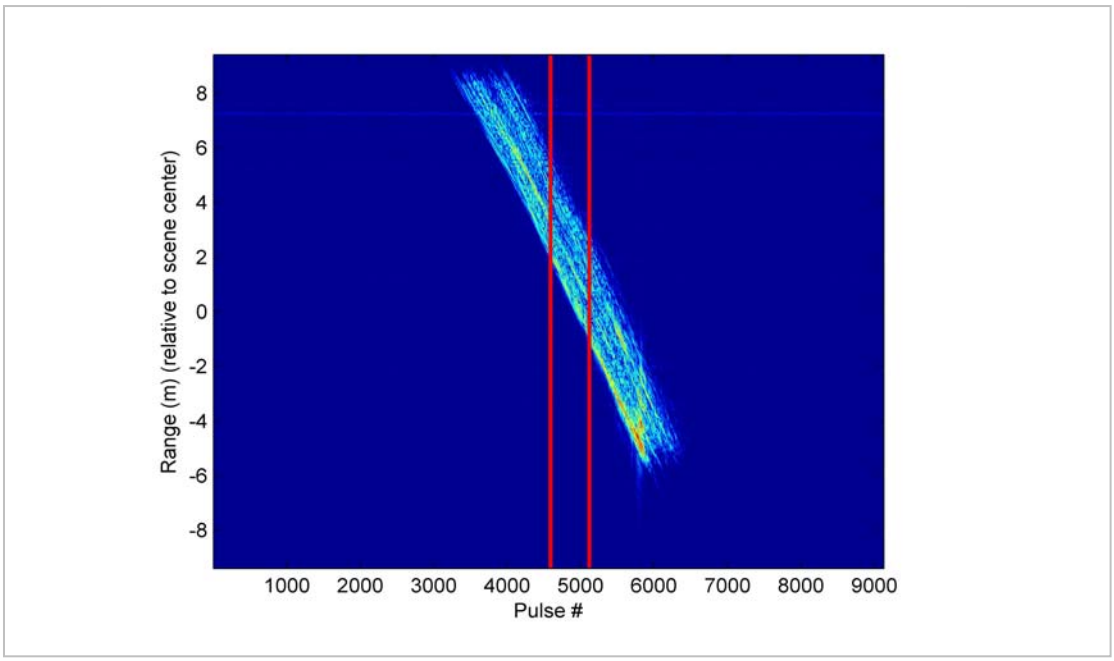


Figure 5-10 Range compressed pulse versus pulse number (vehicle A).

It is evident that the ISAR image formed without motion compensation (Figure 5-11) has much more distortion than the one with motion compensation (Figure 5-12). Even with motion compensation, the ISAR image formed is expected to be defocused because of the uncompensated range (phase) error as well as the rotational or azimuth error.

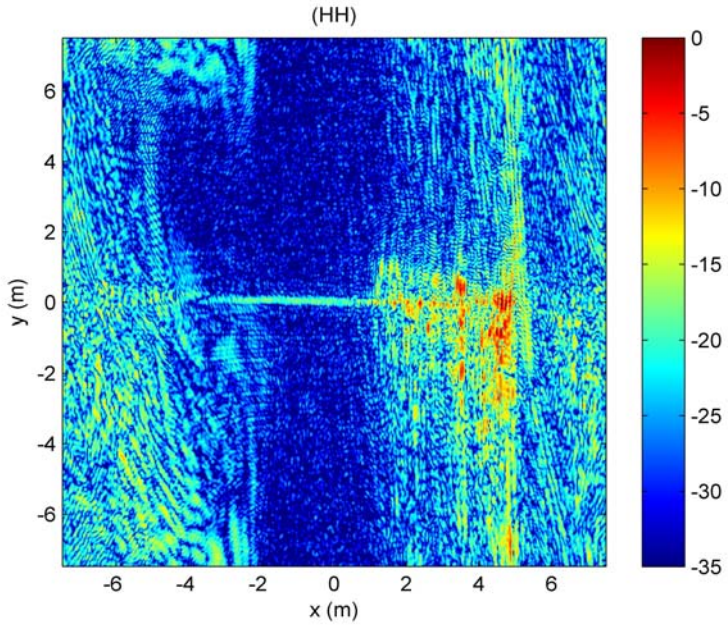


Figure 5-11 Non motion-compensated ISAR image before autofocus (vehicle A).

SCMB is used to focus the motion compensated ISAR image of vehicle A in Figure 5-12. We chose the size of the two-dimensional complex equalization filter that was used within SCMB to be 121x15 in this example, where 121 is the filter support in the cross-range dimension (y axis) and 15 is the support in the range dimension (x axis).

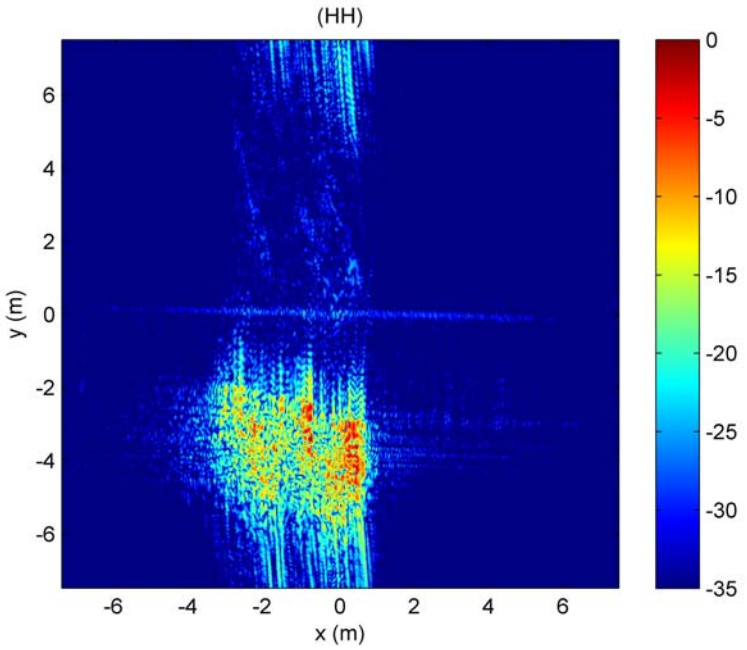


Figure 5-12 Motion-compensated ISAR image before autofocus (vehicle A).

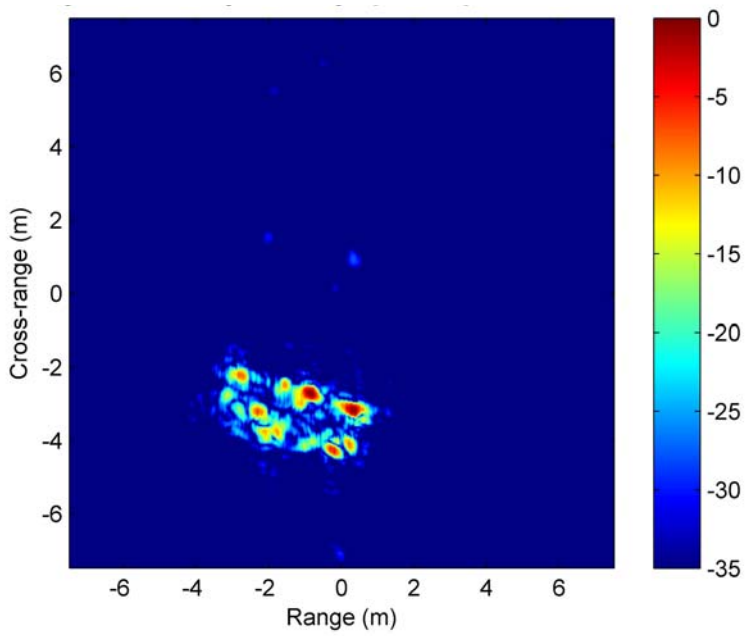


Figure 5-13 ISAR autofocus using SCMB (vehicle A).

The result of the SCMB focused ISAR image of vehicle A is shown in Figure 5-13, where the point scatterers of the vehicle become apparent. For comparison purposes, we apply PGA and minimum entropy autofocus [19, 47, 48] to focus the same motion compensated ISAR image in Figure 5-12 and the results are shown, respectively, in Figure 5-14 and Figure 5-15. PGA evidently performs better than minimum entropy, but both results are inferior to the result of SCMB.

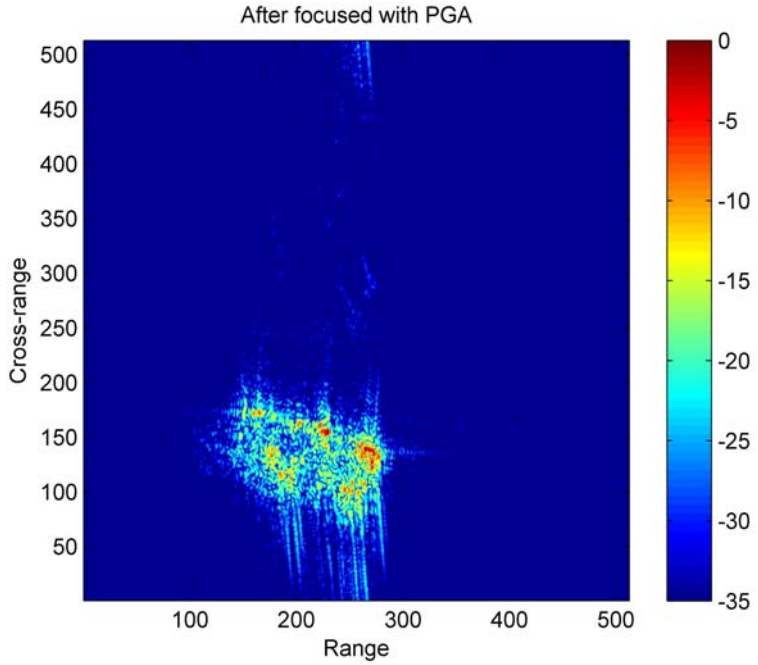


Figure 5-14 ISAR autofocus using phase gradient algorithm (PGA), (vehicle A).

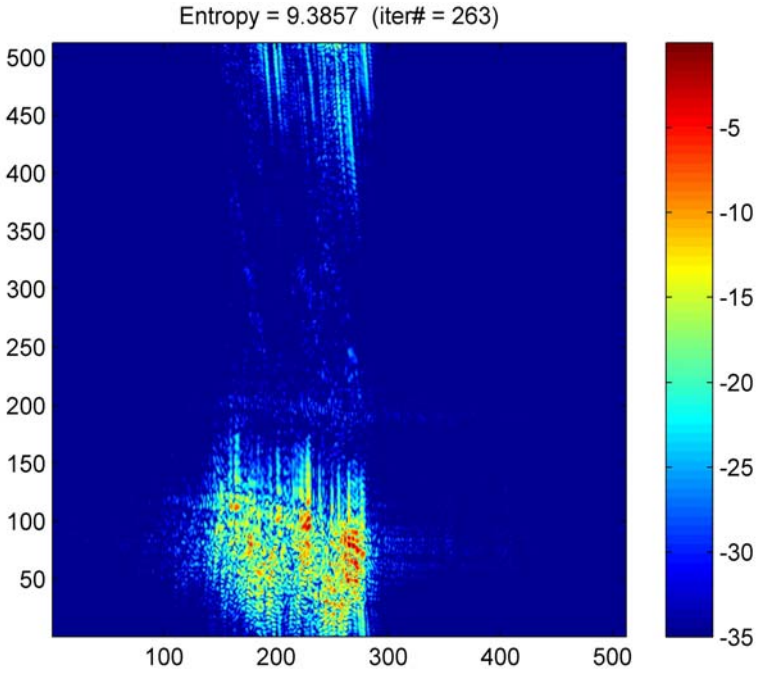


Figure 5-15 ISAR autofocus using minimum entropy approach (vehicle A).

We repeat the experiment as described above for vehicle B for a different radar pass with the center azimuth angle of again roughly 25 degrees with respect to the vehicle's body axis. In Figure 5-16, the range compressed pulse diagram for vehicle B appears to be noisier than that of vehicle A. Since the radar is stationary and the background does not change in both cases, this observation suggests that vehicle A has stronger back scatter, possibly caused by more and larger trihedrals than vehicle B.

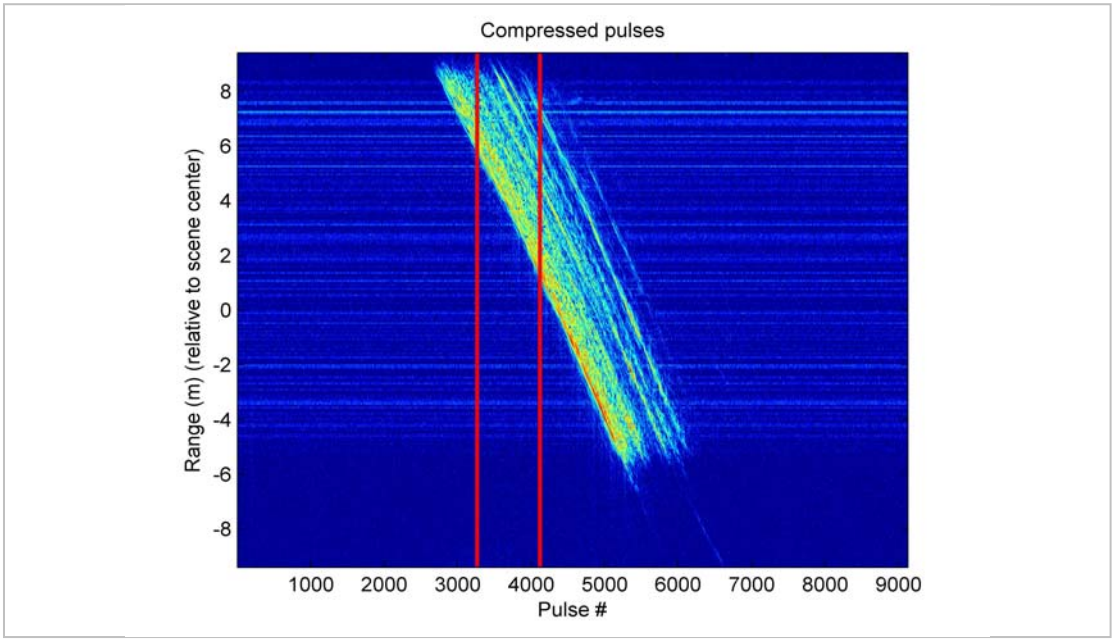


Figure 5-16 Range compressed pulse versus pulse number (vehicle B).

By comparing the two range compressed pulse diagrams, it is also noticeable that vehicle B should be longer than vehicle A in the range dimension with the current orientation. The ISAR image of vehicle B formed without and with motion compensation are shown in Figure 5-17 and Figure 5-18 respectively. It is not surprising that much more distortion is found in the ISAR image formed without motion compensation than in the one with motion compensation.

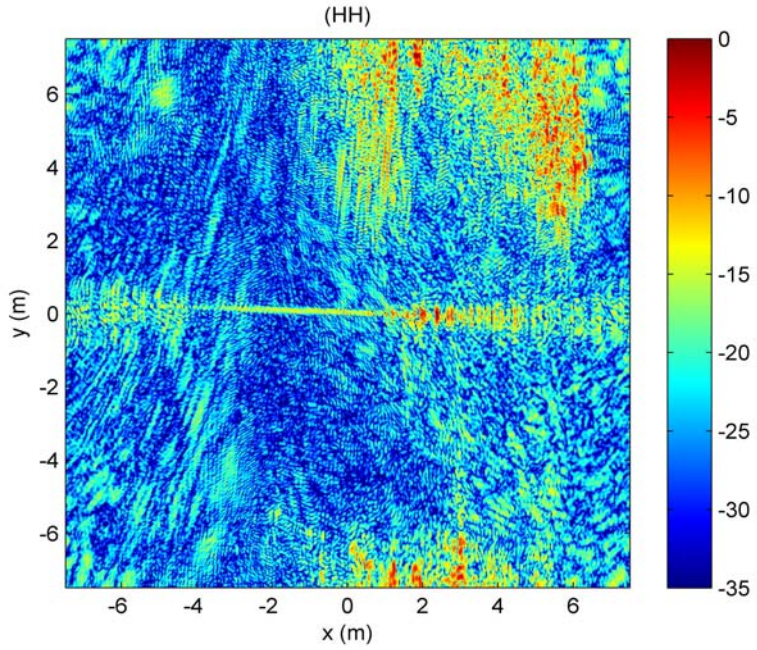


Figure 5-17 Non motion-compensated ISAR image before autofocus (vehicle B).

SCMB was then used on the motion-compensated ISAR image of vehicle B in Figure 5-18. Again, we chose the size of the two-dimensional complex equalization filter that was used within SCMB to be 121x15, where 121 is the support in the cross-range dimension (y axis) and 15 is the support in the range dimension (x axis). The resulting SCMB-focused ISAR image of vehicle B is shown in Figure 5-19. Again the point scatterers of the vehicle become apparent.

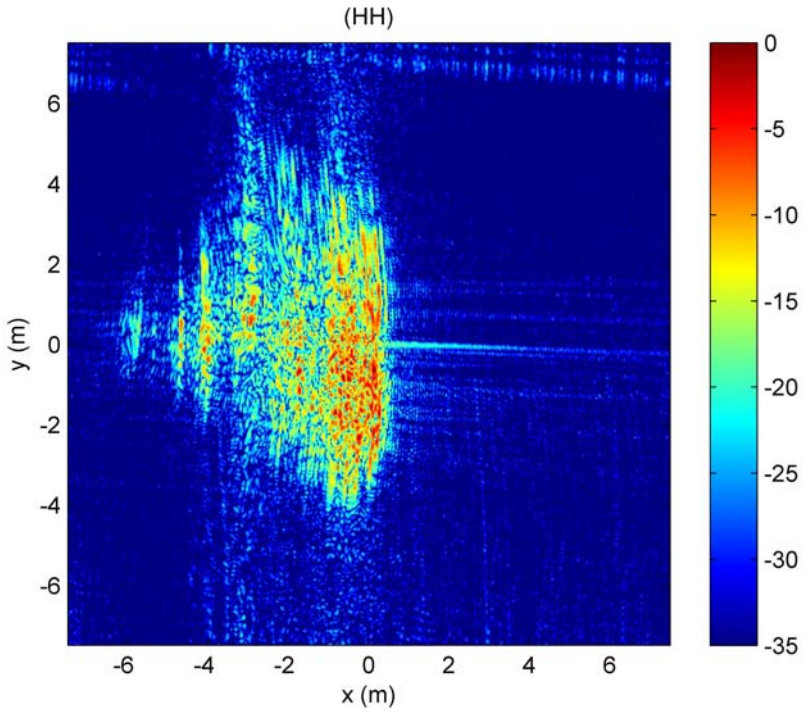


Figure 5-18 Motion-compensated ISAR image before autofocus (vehicle B)

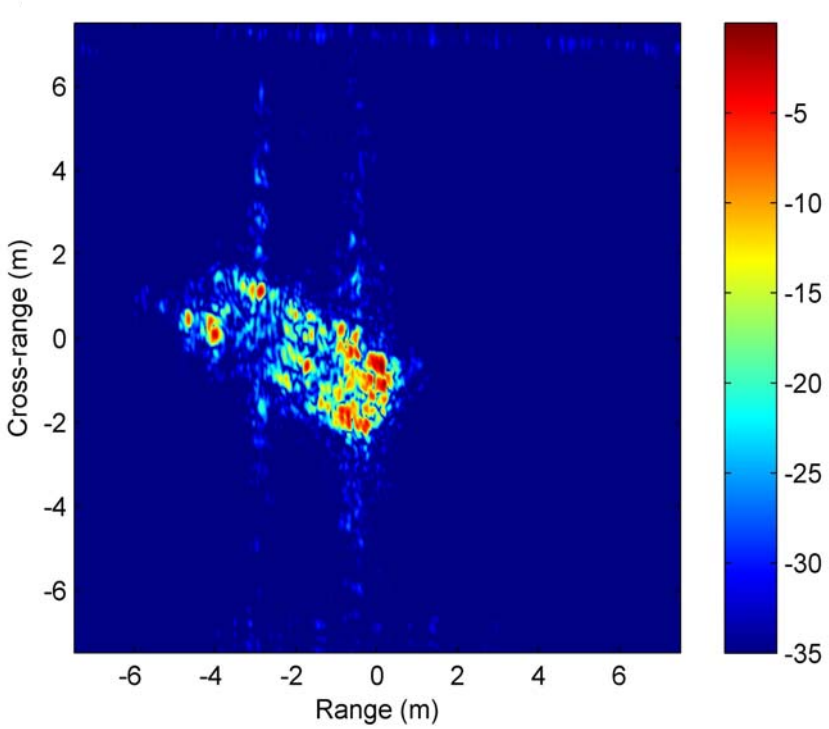


Figure 5-19 ISAR autofocus using SCMB (vehicle B)

We then apply PGA and minimum entropy autofocus to focus the motion compensated ISAR image of vehicle B in Figure 5-18 for comparision. The results are shown, respectively, in Figure 5-20 and **Error! Reference source not found.**. PGA again out-performs minimum entropy, but nevertheless both methods are clearly vastly inferior to the performance of the SCMB blind deconvolution algorithm.

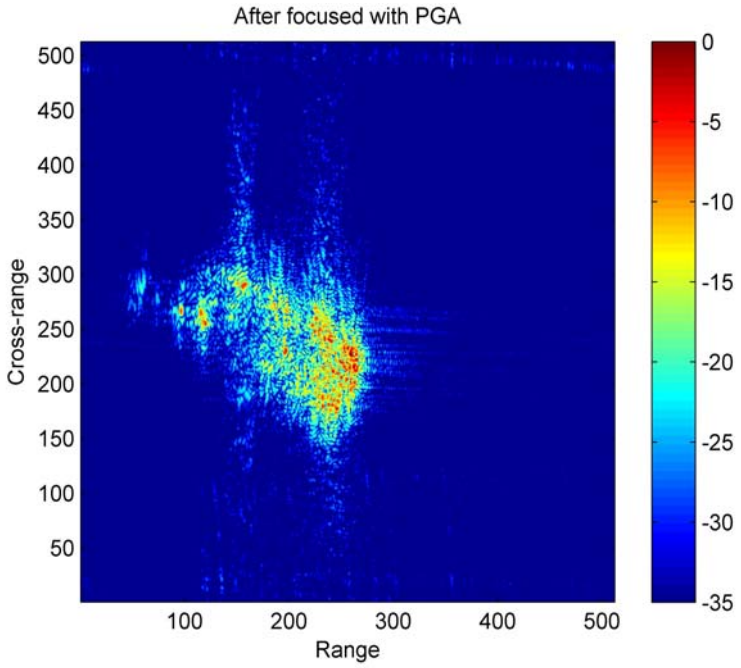


Figure 5-20 ISAR autofocus using phase gradient algorithm (PGA), (vehicle B).

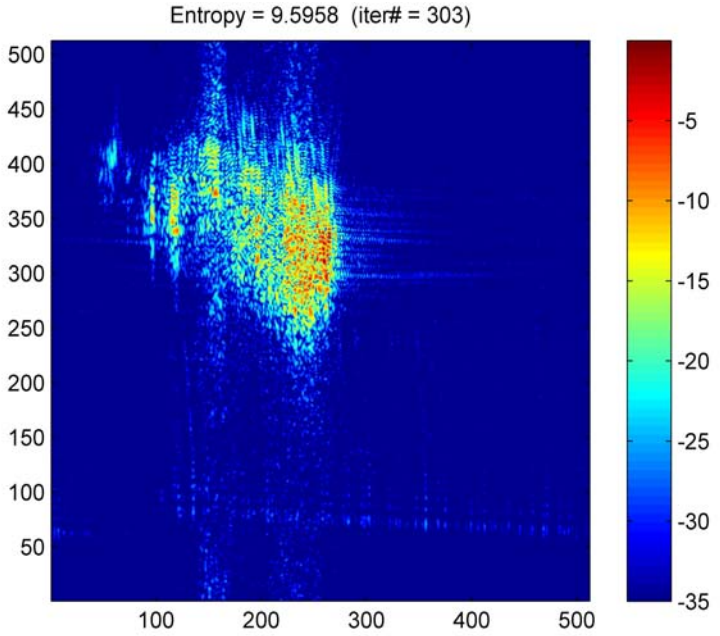


Figure 5-21 ISAR autofocus using minimum entropy approach (vehicle B).

We conclude that the non-uniform turning rate of the vehicle in this experiment makes the rotational (azimuth) error nontrivial comparing to the range (phase) error. Therefore, SAR autofocus techniques like PGA and minimum entropy which do not address the rotational error become quite unsuitable for the ISAR autofocus case when there is significant non-uniform rotational error. Finally, by comparing the ISAR autofocus results using SCMB blind deconvolution for vehicle A and vehicle B, respectively in Figure 5-13 and Figure 5-19, we can clearly distinguish vehicle A from vehicle B by the dimensions of the vehicle, and the characteristics and distribution of the point scatterers. That makes automatic target recognition (ATR) and ground moving target imaging (GMTI) more feasible when this method is used for preprocessing.

6 Conclusion

In this dissertation, we present a unique multi-channel blind deconvolution framework derived from the Bussgang algorithm which is applicable to multiple disciplines. We named the algorithm the self-correcting multi-channel Bussgang (SCMB) blind deconvolution algorithm.

6.1 Contributions

1. A practical binary image restoration technique which does not assume the knowledge of the pdf of the original undistorted binary image.

    - We started our binary image restoration study focusing on Panci's paper in [42] because of its amazing binary image restoration results. However, on closer study, we found that the pdf of the original binary images, which are practically inaccessible, are used to produce those results. We presented more realistic results obtained without knowing the pdf in Figure 3-3. In order to make this Bussgang-based binary image restoration approach work without prior knowledge of the pdf, we developed the SCMB blind deconvolution algorithm described in Section 3.

2. The first known multi-channel binary image restoration technique applicable to severely blurred binary images.

    - This is to our knowledge the first multi-channel binary image restoration algorithm that can utilize multiple blurred binary images and restore severely blurred images as shown in the examples in Figure 4-9 and Figure 4-10.

3. The first known SAR/ISAR autofocus algorithm that can take advantage of multiple similar radar passes when available.

    - Unlike mainstream SAR/ISAR autofocus algorithms, this is the only algorithm that can take advantage of multiple similar radar passes to improve autofocus performance. An example using real data is shown in Figure 5-5

4. An innovative SAR/ISAR autofocus technique not based on current popular techniques such as prominent scatterer tracking, minimum entropy and phase gradient algorithm (PGA) autofocus. Our algorithm was shown to out-perform PGA and minimum entropy.

- A comparision with Ku-band radar data between our SCMB blind deconvolution algorithm for SAR autofocus and PGA is shown in Figure 5-5, Figure 5-6 and Figure 5-7.

5. The first known ISAR autofocus technique validated with real data for ground moving target imaging under significant translational and rotational motion.

    - Results for ISAR autofocus that are validated with real data are quite uncommon. The only ISAR autofocus result that deals with real data, to our knowledge, is to focus a navy ship in linear motion with some mild rolling [35]. However, the defocusing was mild in the particular case presented.

6. A unified approach for both SAR and ISAR autofocus.

    - Our SCMB blind deconvlution algorithm for SAR/ISAR autofocus is effective for both the SAR and ISAR autofocus problems. A 1D equalization filter support in the cross-range dimension is usually used for the SAR autofocus problem while a 2D equalization filter support is used in the ISAR autofocus problem, but the algorithm is the same in either case.

6.2 Suggestions for future work

1. Exploring applications for the binary image restoration technique in text printing, computer vision, or in communication with unknown or time-varying statistics.

    - The SCMB blind deconvolution framework for binary image restoration is a crucial step in our research for concept verification. However, unlike SAR/ISAR autofocus, its application is limited. Therefore, we would like to exploit any potential applications that can take advantage of the technique.

2. Extending the technique to three-dimensional SAR formation with k-space sparsity.

    - For multiple radar passes with different depression angles it is possible to form a three-dimensional SAR image or volume using all passes simultaneously only if we can conquer the aliasing problem caused by k-space sparsity, that is, not enough passes to cover all the depression angles needed for the required dimensions of the imaged volume.

3. Extending the technique to photographic image restoration.

    - The SCMB blind deconvolution framework can potentially apply to photographic image restoration from multiple snap-shots to recover bad or blurred images caused by over or under exposure, motion, out-of-focus and more.

4. Investigate the benefit of introducing a spatial constraint into the SCMB blind deconvolution framework such as an AR or ARMA model.

    - The SCMB blind deconvolution framework, like the original Bussgang framework, constraints the solution in a statistical sense in the form of the pdf or the Busggang nonlinearity. It will be interesting to investigate the effect of further constraining the solution using some spatial constraints when available such as an AR or ARMA model.

Appendix

A.1 Partial derivatives for the feedback signal in the SAR/ISAR case

We present the partial derivatives generated by Mathematica for equation (5.30). For completeness, (5.30) is restated here:

$$\Delta\hat{z} = -\sum_{i=1}^{3}\left\{ \frac{\left[\frac{\partial \hat{z}}{\partial w_i}\right]}{\left[\frac{\partial \hat{z}}{\partial \tilde{z}}\right]}\Delta w_i + \frac{\left[\frac{\partial z}{\partial \mu_i}\right]}{\left[\frac{\partial z}{\partial \tilde{z}}\right]}\Delta\mu_i + \frac{\left[\frac{\partial z}{\partial \sigma_i}\right]}{\left[\frac{\partial z}{\partial \tilde{z}}\right]}\Delta\sigma_i \right\}. \tag{7.1}$$

In an actual implementation, it is more compact to have the feedback signal depend on the variances of the Gaussian components instead of their standard deviations. Therefore, the feedback signal, $\Delta\hat{z}$, can be rewritten as

$$\Delta\hat{z} = -\sum_{i=1}^{3}\left\{ \frac{\left[\frac{\partial \hat{z}}{\partial w_i}\right]}{\left[\frac{\partial \hat{z}}{\partial \tilde{z}}\right]}\Delta w_i + \frac{\left[\frac{\partial z}{\partial \mu_i}\right]}{\left[\frac{\partial z}{\partial \tilde{z}}\right]}\Delta\mu_i + \frac{\left[\frac{\partial \hat{z}}{\partial \nu_p(i)}\right]}{\left[\frac{\partial z}{\partial \tilde{z}}\right]}\Delta\nu_p(i) \right\}, \tag{7.2}$$

where $\nu_p(i)$ is the variance of the $i-th$ component of the Gaussian mixture.

We also rewrite the Bussgang nonlinearity output, $\hat{z}$, for SAR/ISAR in (5.20) here for completeness,

$$\hat{z} = \frac{\sum_{i=1}^{3}\left(a_i \frac{w_i\, e^{-\left(\frac{1}{2}\right)d_i}}{\sqrt{2\pi\left({\sigma_{w_z}}^2 + {\sigma_i}^2\right)}} \right)}{\sum_{i=1}^{3}\left(\frac{w_i\, e^{-\left(\frac{1}{2}\right)d_i}}{\sqrt{2\pi\left({\sigma_{w_z}}^2 + {\sigma_i}^2\right)}} \right)}, \tag{7.3}$$

where $a_i = \frac{{\sigma_{w_z}}^2\mu_i + {\sigma_i}^2\hat{z}}{{\sigma_{w_z}}^2 + {\sigma_i}^2}$, $d_i = \frac{\left(\hat{z}-\mu_i\right)^2}{{\sigma_{w_z}}^2 + {\sigma_i}^2}$ and σ_{w_z} is the standard deviation of the deconvolution noise for the magnitude of the input signal.

To simplify the expression for the derivatives in (7.2), we define the variables, $T1(i)$, $T2(i)$, $T3(i)$, $T4(i)$ and $s(i)$ as follows:

$$T1(i) = \hat{z} - \mu(i) \tag{7.4}$$

$$T2(i) = \nu_p(i) - \sigma_{w_z}^2 \tag{7.5}$$

$$T3(i) = \mu(i)\sigma_{w_z}^2 + \hat{z}T2(i) \tag{7.6}$$

$$T4(i) = e^{-\frac{T1(i)^2}{2\nu_p(i)}} \tag{7.7}$$

$$s(i) = \sqrt{T2(i)} \tag{7.8}$$

By taking the derivatives of equation (7.2) with respect to w_i, μ_i, $\nu_p(i)$ and $\hat{z}$ using Mathematica, we have the results in the following table in terms of $T1(i)$, $T2(i)$, $T3(i)$, $T4(i)$ and $s(i)$.

	$$\frac{\partial \hat{z}}{\partial v_p(i)} = \frac{\mathrm{T4(i)}w(i)\left(\begin{array}{l}\left(\sigma_{\mathrm{wz}}^2\,\mathrm{T1(i)}^3 - \left(\hat{z}^2 - \mu(i)\hat{z} + 3\sigma_{\mathrm{wz}}^2\right)v_p(i)T1(\mathrm{i}) + \hat{z}v_p(i)^2\right)\left(\frac{\mathrm{T4(1)}w(1)}{\sqrt{v_p(1)}} + \frac{\mathrm{T4(2)}w(2)}{\sqrt{v_p(2)}} + \frac{\mathrm{T4(3)}w(3)}{\sqrt{v_p(3)}}\right) + \left(\mathrm{T1(i)}^2 - v_p(i)\right)v_p(i) \\ \left(\frac{\mathrm{T4(1)}w(1)\left(\hat{z}v_p(1) - T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}} + \frac{\mathrm{T4(2)}w(2)\left(\hat{z}v_p(2) - T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}} + \frac{\mathrm{T4(3)}w(3)\left(\hat{z}v_p(3) - T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)\end{array}\right)}{2v_p(i)^{7/2}\left(\frac{\mathrm{T4(1)}w(1)\left(\hat{z}v_p(1) - T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}} + \frac{\mathrm{T4(2)}w(2)\left(\hat{z}v_p(2) - T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}} + \frac{\mathrm{T4(3)}w(3)\left(\hat{z}v_p(3) - T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)^2}$$
	$$\frac{\partial \hat{z}}{\partial w(1)} = \frac{e^{\frac{1}{2}\left(-\frac{\mathrm{T1(1)}^2}{v_p(1)} - \frac{\mathrm{T1(2)}^2}{v_p(2)} - \frac{\mathrm{T1(3)}^2}{v_p(3)}\right)}\sigma_{\mathrm{wz}}^2\left(e^{\frac{T1(2)^2}{2v_p(2)}}w(3)v_p(2)^{3/2}\left(\mathrm{T1(1)}v_p(3) - T1(3)v_p(1)\right) - e^{\frac{T1(3)^2}{2v_p(3)}}w(2)\left(\mathrm{T1(2)}v_p(1) - T1(1)v_p(2)\right)v_p(3)^{3/2}\right)}{v_p(1)^{3/2}v_p(2)^{3/2}v_p(3)^{3/2}\left(\frac{\mathrm{T4(1)}w(1)\left(\hat{z}v_p(1) - T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}} + \frac{\mathrm{T4(2)}w(2)\left(\hat{z}v_p(2) - \mathrm{t12}\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}} + \frac{\mathrm{T4(3)}w(3)\left(\hat{z}v_p(3) - T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)^2}$$

Table 6-1 Partial derivatives for the feedback signal in the SAR/ISAR case generated by Mathematica.

	$$\frac{\partial \hat{z}}{\partial v_p(i)} = \frac{\mathrm{T4(i)}w(i)\left(\begin{array}{l}\left(\sigma_{\mathrm{wz}}^2\,\mathrm{T1(i)}^3 - \left(\hat{z}^2 - \mu(i)\hat{z} + 3\sigma_{\mathrm{wz}}^2\right)v_p(i)T1(\mathrm{i}) + \hat{z}v_p(i)^2\right)\left(\frac{\mathrm{T4(1)}w(1)}{\sqrt{v_p(1)}} + \frac{\mathrm{T4(2)}w(2)}{\sqrt{v_p(2)}} + \frac{\mathrm{T4(3)}w(3)}{\sqrt{v_p(3)}}\right) + \left(\mathrm{T1(i)}^2 - v_p(i)\right)v_p(i) \\ \left(\frac{\mathrm{T4(1)}w(1)\left(\hat{z}v_p(1) - T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}} + \frac{\mathrm{T4(2)}w(2)\left(\hat{z}v_p(2) - T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}} + \frac{\mathrm{T4(3)}w(3)\left(\hat{z}v_p(3) - T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)\end{array}\right)}{2v_p(i)^{7/2}\left(\frac{\mathrm{T4(1)}w(1)\left(\hat{z}v_p(1) - T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}} + \frac{\mathrm{T4(2)}w(2)\left(\hat{z}v_p(2) - T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}} + \frac{\mathrm{T4(3)}w(3)\left(\hat{z}v_p(3) - T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)^2}$$
	$$\frac{\partial \hat{z}}{\partial w(1)} = \frac{e^{\frac{1}{2}\left(-\frac{\mathrm{T1(1)}^2}{v_p(1)} - \frac{\mathrm{T1(2)}^2}{v_p(2)} - \frac{\mathrm{T1(3)}^2}{v_p(3)}\right)}\sigma_{\mathrm{wz}}^2\left(e^{\frac{T1(2)^2}{2v_p(2)}}w(3)v_p(2)^{3/2}\left(\mathrm{T1(1)}v_p(3) - T1(3)v_p(1)\right) - e^{\frac{T1(3)^2}{2v_p(3)}}w(2)\left(\mathrm{T1(2)}v_p(1) - T1(1)v_p(2)\right)v_p(3)^{3/2}\right)}{v_p(1)^{3/2}v_p(2)^{3/2}v_p(3)^{3/2}\left(\frac{\mathrm{T4(1)}w(1)\left(\hat{z}v_p(1) - T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}} + \frac{\mathrm{T4(2)}w(2)\left(\hat{z}v_p(2) - \mathrm{t12}\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}} + \frac{\mathrm{T4(3)}w(3)\left(\hat{z}v_p(3) - T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)^2}$$

Table 6-2 Partial derivatives for the feedback signal in the SAR/ISAR case generated by Mathematica (continued).

	$$\frac{\partial \hat{z}}{\partial w(2)} = \frac{e^{\frac{1}{2}\left(-\frac{\mathrm{T1}(1)^2}{v_p(1)}-\frac{\mathrm{T1}(2)^2}{v_p(2)}-\frac{\mathrm{T1}(3)^2}{v_p(3)}\right)}\sigma_{\mathrm{wz}}^2\left(e^{\frac{T1(1)^2}{2v_p(1)}}w(3)v_p(1)^{3/2}\left(\mathrm{T1}(2)v_p(3)-T1(3)v_p(2)\right)-e^{\frac{T1(3)^2}{2v_p(3)}}w(1)\left(\mathrm{T1}(1)v_p(2)-T1(2)v_p(1)\right)v_p(3)^{3/2}\right)}{v_p(1)^{3/2}v_p(2)^{3/2}v_p(3)^{3/2}\left(\frac{\mathrm{T4}(1)w(1)\left(\hat{z}v_p(1)-T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}}+\frac{\mathrm{T4}(2)w(2)\left(\hat{z}v_p(2)-T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}}+\frac{\mathrm{T4}(3)w(3)\left(\hat{z}v_p(3)-T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)^2}$$
	$$\frac{\partial \hat{z}}{\partial w(3)} = \frac{e^{\frac{1}{2}\left(-\frac{\mathrm{T1}(1)^2}{v_p(1)}-\frac{\mathrm{T1}(2)^2}{v_p(2)}-\frac{\mathrm{T1}(3)^2}{v_p(3)}\right)}\sigma_{\mathrm{wz}}^2\left(e^{\frac{T1(1)^2}{2v_p(1)}}w(2)v_p(1)^{3/2}\left(\mathrm{T1}(3)v_p(2)-T1(2)v_p(3)\right)-e^{\frac{T1(2)^2}{2v_p(2)}}w(1)\left(\mathrm{T1}(1)v_p(3)-T1(3)v_p(1)\right)v_p(2)^{3/2}\right)}{v_p(1)^{3/2}v_p(2)^{3/2}v_p(3)^{3/2}\left(\frac{\mathrm{T4}(1)w(1)\left(\hat{z}v_p(1)-T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}}+\frac{\mathrm{T4}(2)w(2)\left(\hat{z}v_p(2)-T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}}+\frac{\mathrm{T4}(3)w(3)\left(\hat{z}v_p(3)-T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)^2}$$

Table 6-3 Partial derivatives for the feedback signal in the SAR/ISAR case generated by Mathematica (continued).

	$$\frac{\partial \hat{z}}{\partial \mu(i)} = \frac{\mathrm{T4(i)}w(i)\left(\begin{array}{l}\mathrm{T1(i)}v_p(i)\left(\frac{\mathrm{T4}(1)w(1)\left(\hat{z}v_p(1)-T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}}+\frac{\mathrm{T4}(2)w(2)\left(\hat{z}v_p(2)-T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}}+\frac{\mathrm{T4}(3)w(3)\left(\hat{z}v_p(3)-T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)\\ -\left(\left(\hat{z}^2-\mu(i)\hat{z}+\sigma_{\mathrm{wz}}^2\right)v_p(i)-T1(\mathrm{i})^2\sigma_{\mathrm{wz}}^2\right)\left(\frac{\mathrm{T4}(1)w(1)}{\sqrt{v_p(1)}}+\frac{\mathrm{T4}(2)w(2)}{\sqrt{v_p(2)}}+\frac{\mathrm{T4}(3)w(3)}{\sqrt{v_p(3)}}\right)\end{array}\right)}{v_p(1)^{5/2}\left(\frac{\mathrm{T4}(1)w(1)\left(\hat{z}v_p(1)-T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}}+\frac{\mathrm{T4}(2)w(2)\left(\hat{z}v_p(2)-T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}}+\frac{\mathrm{T4}(3)w(3)\left(\hat{z}v_p(3)-T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)^2}$$
	$$\frac{\partial \hat{z}}{\partial \bar{z}} = \frac{\left(-\frac{\mathrm{T1}(1)\mathrm{T4}(1)w(1)}{v_p(1)^{3/2}}-\frac{\mathrm{T1}(2)\mathrm{T4}(2)w(2)}{v_p(2)^{3/2}}-\frac{\mathrm{T1}(3)\mathrm{T4}(3)w(3)}{v_p(3)^{3/2}}\right)\left(\frac{\mathrm{T4}(1)w(1)\left(\hat{z}v_p(1)-T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}}+\frac{\mathrm{T4}(2)w(2)\left(zv_p(2)-T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}}+\frac{\mathrm{T4}(3)w(3)\left(zv_p(3)-T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)}{\left(\frac{\mathrm{T4}(1)w(1)\left(\hat{z}v_p(1)-T1(1)\sigma_{\mathrm{wz}}^2\right)}{v_p(1)^{3/2}}+\frac{\mathrm{T4}(2)w(2)\left(zv_p(2)-T1(2)\sigma_{\mathrm{wz}}^2\right)}{v_p(2)^{3/2}}+\frac{\mathrm{T4}(3)w(3)\left(zv_p(3)-T1(3)\sigma_{\mathrm{wz}}^2\right)}{v_p(3)^{3/2}}\right)^2}$$ $$-\left(\frac{\mathrm{T4}(1)w(1)}{\sqrt{v_p(1)}}+\frac{\mathrm{T4}(2)w(2)}{\sqrt{v_p(2)}}+\frac{\mathrm{T4}(3)w(3)}{\sqrt{v_p(3)}}\right)\left(\begin{array}{l}\frac{\mathrm{T1}(1)\mathrm{T4}(1)\left(\mathrm{T1}(1)\sigma_{\mathrm{wz}}^2-\hat{z}v_p(1)\right)w(1)}{v_p(1)^{5/2}}+\frac{\mathrm{T2}(1)\mathrm{T4}(1)w(1)}{v_p(1)^{3/2}}+\frac{\mathrm{T1}(2)\mathrm{T4}(2)w(2)\left(\mathrm{T1}(2)\sigma_{\mathrm{wz}}^2-zv_p(2)\right)}{v_p(2)^{5/2}}+\\ \frac{\mathrm{T1}(3)\mathrm{T4}(3)w(3)\left(\mathrm{T1}(3)\sigma_{\mathrm{wz}}^2-\hat{z}v_p(3)\right)}{v_p(3)^{5/2}}+\frac{\mathrm{T2}(2)\mathrm{T4}(2)w(2)}{v_p(2)^{3/2}}+\frac{\mathrm{T2}(3)\mathrm{T4}(3)w(3)}{v_p(3)^{3/2}}\end{array}\right)$$

Table 6-4 Partial derivatives for the feedback signal in the SAR/ISAR case generated by Mathematica (continued).

A.2 Derivation of the Bussgang nonlinearity w.r.t. pdf

The Bussgang nonlinearity w.r.t. the pdf of the input signal and the deconvolution noise in (3.11),

$$g\{\breve{x}\} \triangleq E\{x \mid \breve{x}\} = \frac{\int x\, p_x(x)\, p_w(\breve{x}-x)\, dx}{\int p_x(x)\, p_w(\breve{x}-x)\, dx},$$

is derived here.

The Bayes' risk defined in (2.12) is rewritten as

$$J(\hat{x}) = \iint C(x - \hat{x}(\breve{x}))\; p_{x,\breve{x}}(x,\breve{x})\, dx\, d\breve{x}.$$

By choosing the Bussgang cost function $C(x-\hat{x}(\breve{x}))$ as the $l_2-\text{norm}$, the Bayes' risk becomes the mean square error (MSE) and is given by

$$MSE = \iint (x - \hat{x}(\breve{x}))^T (x - \hat{x}(\breve{x}))\; p_{x,\breve{x}}(x,\breve{x})\, dx\, d\breve{x}. \tag{7.9}$$

The nonlinearity output $\hat{x} \triangleq g\{\breve{x}\}$ is chosen to minimize the MSE in (7.9). Therefore, we have

$$\frac{\partial MSE}{\partial \hat{x}} = \int\int -2\big(x-\hat{x}(\breve{x})\big) p_{x,\breve{x}}(x,\breve{x}) dx d\breve{x} \triangleq 0. \tag{7.10}$$

The above equation can be written as

$$\iint \hat{x}(\breve{x})\, p_{x,\breve{x}}(x,\breve{x}) dx d\breve{x} = \iint x\, p_{x,\breve{x}}(x,\breve{x}) dx d\breve{x}. \tag{7.11}$$

By reorganizing the integral on the left-hand side and applying the Bayes' rule on the right-hand side, we have

$$\int \hat{x}(\breve{x}) \Big[\int p_{x,\breve{x}}(x,\breve{x})\, dx \Big] d\breve{x} = \iint x\, p_{x|\breve{x}}(x \mid \breve{x})\, p_{\breve{x}}(\breve{x})\, dx d\breve{x}. \tag{7.12}$$

Again, by reorganizing the right-hand side of (7.12), it becomes

$$\int \hat{x}(\breve{x}) \Big[\int p_{x,\breve{x}}(x,\breve{x})\, dx \Big] d\breve{x} = \int \Big[\int x\, p_{x|\breve{x}}(x \mid \breve{x})\, dx \Big] p_{\breve{x}}(\breve{x})\, d\breve{x}. \tag{7.13}$$

Now, by evaluating the expressions inside the square brackets on both side of (7.13), it can be more compactly written as

$$\int \hat{x}(\breve{x})\, p_{\breve{x}}(\breve{x})\, d\breve{x} = \int E\{x \mid \breve{x}\}\, p_{\breve{x}}(\breve{x})\, d\breve{x}\,. \tag{7.14}$$

As a result, one feasible solution for (7.14) is to have the nonlinearity $\hat{x}$ equal to the conditional expectation of x given the deconvolved output $\breve{x}$. That is,

$$\hat{x} = E\{x \mid \breve{x}\}\,. \tag{7.15}$$

We can write the conditional expectation in term of integral and then apply Bayes' rule, giving

$$E\{x \mid \breve{x}\} = \int x\, p_{x|\breve{x}}(x \mid \breve{x})\, dx = \frac{\int x\, p_{x,\breve{x}}(x, \breve{x}) dx}{p_{\breve{x}}(\breve{x})}\,. \tag{7.16}$$

From the definition of the deconvolution noise $w = \breve{x} - x$, we have the deconvolved output $\breve{x}$ expressed as

$$\breve{x} = w + x\,. \tag{7.17}$$

The pdf of the sum of two independent random variables is equal to the convolution of the pdf of the two random variables. That is,

$$p_{\breve{x}}(\breve{x}) = p_w(\breve{x}) * p_x(\breve{x}) = \int p_x(x)\, p_w(\breve{x} - x) dx\,. \tag{7.18}$$

Notice also that

$$p_{x,\breve{x}}(x, \breve{x}) = p_{x,w}(x, w) = p_x(x)\, p_w(w) = p_x(x)\, p_w(\breve{x} - x)\,, \tag{7.19}$$

since x and w are independent. By substituting (7.18) and (7.19) into (7.16), the conditional expectation becomes

$$E\{x \mid \breve{x}\} = \frac{\int x\, p_x(x)\, p_w(\breve{x} - x)\, dx}{\int p_x(x)\, p_w(\breve{x} - x) dx}\,. \tag{7.20}$$

This concludes our derivation of the Bussgang nonlinearity in terms of the pdf of the input signal and the deconvolution noise.

A.3 Derivation of the identities used in SAR phase nonlinearity

Proof of the identity in (5.26) as rewritten below,

$$\int_{-\pi}^{\pi} u\, e^{-\frac{(a-u)^2}{b^2}}\, du = \left(\frac{\sqrt{\pi}}{2}\right) b\, a\left[erf\left(\frac{a+\pi}{b}\right) - erf\left(\frac{a-\pi}{b}\right)\right] + \frac{b^2}{2}\left[e^{-\left(\frac{a+\pi}{b}\right)^2} - e^{-\left(\frac{a-\pi}{b}\right)^2}\right],$$

is given here.

First let $t = \frac{a-u}{b}$ which implies $dt = -\frac{du}{b}$ and by a change of variable from u to t in the left-hand side of the equation above, we have

$$\int_{-\pi}^{\pi} u\, e^{-\frac{(a-u)^2}{b^2}}\, du = -b\int_{\frac{a+\pi}{b}}^{\frac{a-\pi}{b}} (a - b\,t)\, e^{-t^2}\, dt\,. \tag{7.21}$$

Now, by separating the right-hand side of (7.21) into two terms, it becomes

$$\int_{-\pi}^{\pi} u\, e^{-\frac{(a-u)^2}{b^2}}\, du = b\, a\int_{\frac{a-\pi}{b}}^{\frac{a+\pi}{b}} e^{-t^2}\, dt - b^2\int_{\frac{a-\pi}{b}}^{\frac{a+\pi}{b}} t\, e^{-t^2}\, dt\,. \tag{7.22}$$

Now break down the right-hand side of each term of (7.22) into two terms so that the lower limit of the integral is zero and apply the definition of the error function, $erf(u) = \frac{2}{\sqrt{\pi}}\int_0^u e^{-t^2}\, dt$. The resulting expression is given by

$$\int_{-\pi}^{\pi} u\, e^{-\frac{(a-u)^2}{b^2}}\, du = b\, a\left(\frac{\sqrt{\pi}}{2}\right)\left[erf\left(\frac{a+\pi}{b}\right) - erf\left(\frac{a-\pi}{b}\right)\right] - b^2\left[\int_0^{\frac{a+\pi}{b}} t\, e^{-t^2}\, dt - \int_0^{\frac{a-\pi}{b}} t\, e^{-t^2}\, dt\right]. \tag{7.23}$$

In order to get rid of the integrals in (7.23) of the form $\int_0^c t\, e^{-t^2}\, dt$, we use the relation $dt^2 = 2\,t\, dt$ as follows:

$$\int_0^c t\, e^{-t^2}\, dt = \frac{1}{2}\int_0^c e^{-t^2}\, dt^2 = -\frac{1}{2}e^{-t^2}\Big|_0^c = -\frac{1}{2}\left[e^{-c^2} - 1\right]. \tag{7.24}$$

By substituting (7.24) into (7.23) and making use of the fact that the error function is an odd function, we conclude that

$$\int_{-\pi}^{\pi} u\, e^{-\frac{(a-u)^2}{b^2}}\, du = \left(\frac{\sqrt{\pi}}{2}\right) b\, a\left[erf\left(\frac{a+\pi}{b}\right) - erf\left(\frac{a-\pi}{b}\right)\right] + \frac{b^2}{2}\left[e^{-\left(\frac{a+\pi}{b}\right)^2} - e^{-\left(\frac{a-\pi}{b}\right)^2}\right]. \qquad (7.25)$$

This concludes the proof of the identity in (5.26).

Proof of the identity in (5.27) as rewritten below,

$$\int_{-\pi}^{\pi} e^{-\frac{(a-u)^2}{b^2}}\, du = \left(\frac{\sqrt{\pi}}{2}\right) b\left[erf\left(\frac{a+\pi}{b}\right) - erf\left(\frac{a-\pi}{b}\right)\right],$$

is given here.

First let $t = \frac{a-u}{b}$ which implies $dt = -\frac{du}{b}$, and by a change of variable from u to t in the left-hand side of the equation above, we have

$$\int_{-\pi}^{\pi} e^{-\frac{(a-u)^2}{b^2}}\, du = -b\int_{\frac{a+\pi}{b}}^{\frac{a-\pi}{b}} e^{-t^2}\, dt\,. \qquad (7.26)$$

Now break down the right-hand side of (7.26) into two terms, so that the lower limit of the integral is zero,

$$\int_{-\pi}^{\pi} e^{-\frac{(a-u)^2}{b^2}}\, du = -b\left[\int_{0}^{\frac{a-\pi}{b}} e^{-t^2}\, dt - \int_{0}^{\frac{a+\pi}{b}} e^{-t^2}\, dt\right]. \qquad (7.27)$$

This can be rearranged as

$$\int_{-\pi}^{\pi} e^{-\frac{(a-u)^2}{b^2}}\, du = -b\left(\frac{\sqrt{\pi}}{2}\right)\left[\frac{2}{\sqrt{\pi}}\int_{0}^{\frac{a-\pi}{b}} e^{-t^2}\, dt - \frac{2}{\sqrt{\pi}}\int_{0}^{\frac{a+\pi}{b}} e^{-t^2}\, dt\right]. \qquad (7.28)$$

By expressing (7.28) in terms of the error function, we conclude the derivation for identity (5.27). That is,

$$\int_{-\pi}^{\pi} e^{-\frac{(a-u)^2}{b^2}}\, du = \left(\frac{\sqrt{\pi}}{2}\right) b\left[erf\left(\frac{a+\pi}{b}\right) - erf\left(\frac{a-\pi}{b}\right)\right]. \qquad (7.29)$$

REFERENCES

[1] S. Bellini, "Bussgang techniques for blind equalization," in *GLOBECOM '86: IEEE Global Telecommunications Conference. Communications Broadening Technology Horizons. Conference Record (Cat. No.86CH2298-9)*, Houston, TX, USA, 1986, pp. 1634-40.

[2] A. Benveniste, M. Goursat, and G. Ruget, "Robust identification of a nonminimum phase system: blind adjustment of a linear equalizer in data communications," *IEEE Transactions on Automatic Control,* vol. 25, pp. 385-99, June 1980.

[3] H. Mathis and S. C. Douglas, "Bussgang blind deconvolution for impulsive signals," *IEEE Transactions on Signal Processing,* vol. 51, pp. 1905-15, 2003.

[4] H. Xu and H. Zheng, "A simple initialization method for Bussgang class blind equalization," *Journal of System Simulation,* vol. 17, pp. 217-19, 2005.

[5] R. L. Lagendijk, J. Biemond, and D. E. Boekee, "Identification and restoration of noisy blurred images using the expectation-maximization algorithm," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 38, pp. 1180-91, 1990.

[6] K. T. Lay and A. K. Katsaggelos, "Image identification and restoration based on the expectation-maximization algorithm," *Optical Engineering,* vol. 29, pp. 436-45, 1990.

[7] A. P. Petropulu and C. L. Nikias, "Signal reconstruction from the phase of the bispectrum," *IEEE Transactions on Signal Processing,* vol. 40, pp. 601-10, 1992.

[8] D. Hatzinakos, "Carrier phase tracking and tricepstrum-based blind equalization," *Computers & Electrical Engineering,* vol. 18, pp. 109-18, 1992.

[9] D. Hatzinakos and C. L. Nikias, "Blind equalization using a tricepstrum-based algorithm," *IEEE Transactions on Communications,* vol. 39, pp. 669-82, 1991.

[10] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation systems," *IEEE Transactions on Communications,* vol. COM-23, pp. 679-82, 1975.

[11] Y. Sato, "Blind equalization and blind sequence estimation," *IEICE Transactions on Communications,* vol. E77-B, pp. 545-56, 1994.

[12] S. Bellini and F. Rocca, "Blind deconvolution: polyspectra or Bussgang techniques?," in *Digital Communications. Proceedings of the Second Tirrenia International Workshop*, Tirrenia, Italy, 1986, pp. 251-63.

[13] P. Campisi, S. Colonnese, G. Panci, and G. Scarano, "Multichannel Bussgang algorithm for blind restoration of natural images," in *Proceedings 2003 International Conference on Image Processing*, Barcelona, Spain, 2003, pp. 985-8.

[14] P. Campisi, A. Neri, and G. Scarano, "Bussgang deconvolution for texture synthesis and classification," in *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis*, Rome, Italy, 2003, pp. 90-5.

[15] R. Godfrey and F. Rocca, "Zero memory nonlinear deconvolution," *Geophysical Prospecting,* vol. 29, pp. 189-228, 1981.

[16] W. Hsiao-Chun and X. Dongxin, "Blind equalization of communication sequences based on optimization of cumulant criteria," in *WCNC 2003. 2003 IEEE Wireless Communications and Networking Conference Record (Cat. No.03TH8659)*, New Orleans, LA, USA, 2003, pp. 618-22.

[17] G. Panci, S. Colonnese, P. Campisi, and G. Scarano, "Blind equalization for correlated input symbols: A Bussgang approach," *IEEE Transactions on Signal Processing,* vol. 53, pp. 1860-9, 2005.

[18] D. Donoho, "On minimum entropy deconvolution," in *In Applied Time Series Analysis II*, D. Findley, Ed. New York: Academic Press, 1981, pp. 556-608.

[19] R. A. Wiggins, "Minimum entropy deconvolution," in *Proceedings of the International Symposium on Computer Aided Seismic Analysis and Discrimination*, Falmouth, MA, USA, 1977, pp. 7A-14.

[20] D. Li, R. M. Mersereau, and S. Simske, "Blur identification based on kurtosis minimization," in *International Conference on Image Processing*, Genova, Italy 2005.

[21] T. G. Stockham, Jr., T. M. Cannon, and R. B. Ingebretsen, "Blind deconvolution through digital signal processing," *Proceedings of the IEEE,* vol. 63, pp. 678-92, 1975.

[22] D. N. Godard, "Self-recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Transactions on Communications,* vol. CM-28, pp. 1867-75, 1980.

[23] P. He, T. T. Tjhung, and L. K. Rasmussen, "Constant modulus algorithm (CMA) for CDMA communications systems," in *IEEE Vehicular Technology Conference*, Ottawa, Can, 1998, pp. 949-953.

[24] V. Sharma and V. Naveen Raj, "Convergence and performance analysis of Godard family of blind equalization algorithms," in *IEEE International Conference on Communications*, Anchorage, AK, United States, 2003, pp. 2577-2582.

[25] H. H. Zeng and L. Tong, "Blind equalization using the constant modulus algorithm," in *International Conference on Signal Processing Proceedings*, Beijing, China, 1996, pp. 400-403.

[26] S. Haykin, "Blind equalization formulated as a self-organized learning process," in *Conference Record of The Twenty-Sixth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 1992, pp. 346-50.

[27] S. S. Haykin, 1931- *Adaptive filter theory / Simon Haykin.*: Upper Saddle River, N.J. : Prentice Hall, c1996.

[28] W. G. Carrara, *Spotlight synthetic aperture radar : signal processing algorithms / Walter G. Carrara, Ron S. Goodman, Ronald M. Majewski.*: Boston : Artech House, c1995.

[29] C. V. Jakowatz, Jr., *Spotlight-mode synthetic aperture radar : a signal processing approach / Charles V. Jakowatz, Jr. ... [et al.]*. Boston : Kluwer Academic Publishers, c1996.

[30] P. H. Eichel, D. C. Ghiglia, C. V. Jakowatz, Jr., G. A. Mastin, L. A. Romero, and D. E. Wahl, "Applications of phase gradient autofocus to aperture synthesis imaging," in *Sixth Multidimensional Signal Processing Workshop*, Pacific Grove, CA, USA, 1989, pp. 57-8.

[31] D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and C. V. Jakowatz, Jr., "Phase gradient autofocus - a robust tool for high resolution SAR phase correction," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 30, pp. 827-835, 1994.

[32] C. V. Jakowatz, Jr., P. H. Eichel, and D. C. Ghiglia, "Autofocus of SAR imagery degraded by ionospheric-induced phase errors," in *Proceedings of the SPIE - The International Society for Optical Engineering*, Orlando, FL, USA, 1989, pp. 46-52.

[33] E. H. Satorius and J. J. Mulligan, "Minimum entropy deconvolution and blind equalisation," *Electronics Letters,* vol. 28, pp. 1534-5, 1992.

[34] Y.-B. Huang, Y.-M. Zheng, and Z. Bao, "The SAR/ISAR autofocus based on the multiple dominant scatterers synthesis," *Journal of Xidian University,* vol. 28, pp. 105-9, 2001.

[35] J. M. Munoz-Ferreras, J. Calvo-Gallego, F. Perez-Martinez, A. Blanco-Del-Campo, A. Asensio-Lopez, and B. P. Dorta-Naranjo, "Motion compensation for ISAR based on the shift-and-convolution algorithm," in *CIE International Conference of Radar Proceedings*, New York, United States, 2006, pp. 366-370.

[36] L. Li and S. Mao, "Analysis on phase gradient algorithm in discrete domain and implementation," *Journal of Beijing University of Aeronautics and Astronautics,* vol. 25, pp. 125-9, 1999.

[37] M. P. G. Otten, "Comparison of SAR autofocus algorithms," in *Conference Proceedings. Military Microwaves '90*, London, UK, 1990, pp. 362-7.

[38] B. D. Rigling and R. L. Moses, "Motion measurement errors and autofocus in bistatic SAR," *IEEE Transactions on Image Processing,* vol. 15, pp. 1008-16, 2006.

[39] W. L. Van Rossum, M. P. G. Otten, and R. J. P. Van Bree, "Extended PGA for range migration algorithms," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 42, pp. 478-88, 2006.

[40] R. Wu and J. Li, "Autofocus and super-resolution synthetic aperture radar image formation," *IEE Proceedings-Radar, Sonar and Navigation,* vol. 147, pp. 217-23, 2000.

[41] S. Bellini, "Blind equalization and deconvolution," in *Proceedings of the SPIE - The International Society for Optical Engineering*, San Diego, CA, USA, 1991, pp. 88-101.

[42] G. Panci, P. Campisi, S. Colonnese, and G. Scarano, "Multichannel blind image deconvolution using the Bussgang algorithm: spatial and multiresolution approaches," *IEEE Transactions on Image Processing,* vol. 12, pp. 1324-37, 2003.

[43] R. Kumar, "A fast algorithm for solving a Toeplitz system of equations," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. ASSP-33, pp. 254-67, 1985.

[44] M. Tummala, "Solution of block Toeplitz equations by matrix iterative methods," in *Sixth Multidimensional Signal Processing Workshop*, Pacific Grove, CA, USA, 1989, p. 159.

[45] G. A. Watson, "An algorithm for the inversion of block matrices of Toeplitz form," *Journal of the Association for Computing Machinery,* vol. 20, pp. 409-15, 1973.

[46] M. Wax and T. Kailath, "Efficient inversion of Toeplitz-block Toeplitz matrix," *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. ASSP-31, pp. 1218-21, 1983.

[47] J. Wang and X. Liu, "SAR Minimum-entropy autofocus using an adaptive-order polynomial model," *IEEE Geoscience and Remote Sensing Letters,* vol. 3, pp. 512-516, 2006.

[48] X.-W. Wu and Z.-D. Zhu, "A novel autofocus algorithm based on minimum entropy criteria for SAR images," *Systems Engineering and Electronics,* vol. 25, pp. 867-9, 2003.